IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 6 April 2023, accepted 15 April 2023, date of publication 26 April 2023, date of current version 19 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3270429

RESEARCH ARTICLE

Multi-Circle Detection Guided by Multimodal Optimization Scheme

JORGE GÁLVEZ, ERIK CUEVAS, DAVID ELIEL BOCANEGRA MICHEL, ALMA RODRÍGUEZ, AND MARCO ANTONIO PÉREZ-CISNEROS
Departamento de Electrónica, Universidad de Guadalajara, Guadalajara 44430, Mexico

Corresponding author: Jorge Gálvez (jorge.galvez@academicos.udg.mx)

ABSTRACT Automatic circle detection is one of the most important construction elements to design more complex industrial image tasks such as target detection, manufacture inspection and process control. Most of the literature concerning multi-circle detection can be categorized under deterministic or stochastic perspectives. Deterministic approaches combine geometric and histogram information to identify circle shapes. However, deterministic techniques are unable to detect circles in presence of noise, shape variance, or occlusion. On the other hand, stochastic methodologies such as metaheuristic algorithms have been also proposed as alternatives to deterministic approaches for circle detection. Although these techniques are in general more robust, they allow detecting only one circle per execution, translating the detection problem into a unimodal optimization problem. Under such conditions, it is necessary to execute multiple times the algorithm to identify all the circles contained in the image. In this paper, the multi-circle detection problem has been reformulated as a multimodal optimization problem. The proposed approach adopts the Multimodal Flower Pollination Algorithm (MFPA) to detect all the circular instances contained in the image. Experimental results suggest that the proposed approach significantly improves the multi-circle detection problem.

INDEX TERMS Circle detection, evolutionary algorithms, flower pollination algorithm (FPA), multimodal optimization.

I. INTRODUCTION

Automatic circle detection problem involves the detection of circular shapes over images and video streams. The importance of automatic circle detection can be presented in several fields on engineering, science, and industrial processes [1]. Circle detection is considered as the elemental construction phase for more complex industrial image processing tasks such as manufacture inspection [2], target detection [3], and process control [4]. Most of the related literature to multi-circle detection problem can be categorized under deterministic and stochastic methodologies. Deterministic approaches commonly consider the use of geometric and histogram information on an image to correctly locate circular shapes. Hough transform methods [5] employs an edge detector based on a parametric representation of circles and a voting mechanism. Histogram-based methods commonly used the peak detection information by filtering and averaging the image histogram.

Deterministic approaches provide interesting characteristics to detect circles over an image. However, most of these techniques present some issues in presence of noise and shape artifacts. To overcome such difficulties, researchers have proposed stochastic methodologies to overcome these issues. In general, stochastic circle detection methods include mechanisms such as random sample consensus [6], randomized Hough transform [7], fuzzy Hough transform [8], and evolutionary optimization algorithms.

In stochastic methods, the obtained solution of each methodology is based on the principle of randomness to mimic the abstraction process of uncertainty in human knowledge. Random processes significantly improve the performance of geometric and histogram-based methods by incorporating noise information on their structures and shape invariance to produce fitted solutions. The use of

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

stochastic methods for circle detection has been employed in many fields. Even when stochastic versions of traditional Hough transform or random sample consensus methods have significantly improved the accuracy of the detection of circle shapes, these techniques present some drawbacks such as local minima stagnation [9], premature convergence [10] and scalability in high dimensionality [11], and real-time implementation [12]. Under such circumstances, researchers have adopted evolutionary algorithms to formulate the circle detection problem as an stochastic optimization problem [13], [14], [15].

Evolutionary optimization algorithms are a scalable alternative so solve complex optimization problems. Evolutionary computing combines the abstraction process of several natural, physical, or social phenomena with algorithmic structures to produce search strategies for global optimization. Some examples of evolutionary methods are categorized considering a bioinspired metaphor. Under this scheme, the Genetic Algorithm (GA) [16], the Artificial Bee Colony (ABC) [17], and the Flower Pollination Algorithm (FPA) [18] are some of the most popular methods. Also, considering a physical approach, the Gravitational Search Algorithm (GS) [19], as well as the Electromagnetism-like Optimization (EMO) [20] method are developed. Additionally, inspired by human social behavior, state-of-art literature exposes the War strategy optimization algorithm (WAR) [21] which is a novel search strategy based on ancient war strategy.

Over the last decade, researchers have adopted evolutionary algorithms to detect circular shapes within an image. In [22], the Teaching Learning Algorithm (TLA) has been proposed to improve the circle detection problem using the gradient information in the search space. A hybrid evolutionary segmentation approach based on Kapur thresholding method and the traditional Hough transform is presented in [23]. In the paper, authors improve the search mechanism of circular shape by combining a segmentation methodology to focus the search procedure into promising search zones in the search space. Then, the Hough transform is applied to detect into the promising zones of the histogram to efficiently detect leucocyte sections in hematological images. Pruthi et al. [24] proposed a segmentation method considering the optimization process of an elliptical model in conjunction with the Glowworm Swarm Optimization algorithm to detect glaucoma. In the paper, the evolutionary method is capable for search the optimal solution of the segmentation process considering the elliptical fitting of the solutions. Ayala-Ramirez et al. [25] proposes the use of GA as a circle shape detector. In the experimental section, the GA is able to detect multiple circular shapes but presents overlapping problems. Dasgupta et al. [14] adopts the Bacterial Foraging Optimization Algorithm to detect multiple circles over images. The proposed method is based on an objective function considering the center locations and radio lengths among candidate circles. A comparison study of recent evolutionary approaches for circular shape detection is presented in [26].

Most of the published literature related to circular shape detection using evolutionary algorithms has attracted the attention of many researchers who combine different approaches to significantly improve circular shape detection over images. However, most of these approaches consider a multi-circle detection problem as a unimodal optimization task. Under this methodology, several suboptimal solutions may lead into false positive detection. Under such perspective, the algorithmic structure of unimodal evolutionary algorithms is not capable to handle multiple optima detection in a single run.

Therefore, from a practical perspective it is desirable to detect and maintain all possible circular shapes during the optimization process. The process of finding multiple optima solutions in a single run is known as multimodal optimization.

In multimodal optimization schemes, several mechanisms to efficiently locate, maintain and register both optimal and sub-optimal solutions during the optimization process are required. Over the last decade, some multimodal optimization algorithms have proposed to adopt multimodality into the structure of their exploration and exploitation stages. These multimodal evolutionary strategies have been developed to operate under the real-world multimodal objective functions to obtain not only the optimal solution but also a set of possible sub-optimal solutions. In this paper, the Multimodal Flower Pollination Algorithm (MFPA) [27] has been adopted the original structure of the Flower Pollination Algorithm (FPA) [18] with the extension of multimodal capabilities to detect optimal and sub-optimal solutions into a multi-circle detection problem.

In this paper, the multi-circle detection problem has been reformulated as a real multimodal optimization problem. That is, at each single iteration, the MFPA method can detect not only the best fitted circle in an image but also all potential circles in an image in a single run. The inherent nature of multi-circle detection problem indicates that the detection problem is not solved only by the fittest circle found but for all the potential solutions may be involved in the optimization process. Under such circumstances, the multi-circle detection problem must be solved by a real multimodal optimization algorithm. The experimental study of the proposed approach indicates that MFPA can detect most of the circles even in presence of noise, overlapping objects and avoiding false positive circles in a single run of the algorithm. Experimental results suggest the superior performance of the MFPA against traditional unimodal optimization algorithms commonly used in multi-circle detection problem.

The structure of the paper is organized as follows: Section III describes the algorithmic structure of the original MFPA. In Section III, the multimodal formulation of multi-circle detection problem is defined. Section IV describes the main parts of the multi-circle detection problem into the evolutionary stages of the MFPA. The experimental study is analyzed in Section V. Finally, In Section VI some conclusions and future search directions are presented.

II. MULTIMODAL FLOWER POLLINATION ALGORITHM (MFPA)

Multimodal Flower Pollination Algorithm (MFPA) is the modified version of the original Flower Pollination Algorithm (FPA) to solve multimodal optimization problems proposed by Galvez et al. [27]. The original FPA methodology is based on the biological process of flower pollination. The pollination process consists by transferring pollen between flowers for their reproduction. This mechanism can be carried out by living organisms such as insects, honeybees, birds, or other animals or by non-living organism such as water and wind.

The computational procedure for the FPA method, consists of simplifying the biological aspects of flower pollination in terms of three stages.

- Global pollination trough Lévy flights. Under this mechanism, living organisms travel through Lévy flights over the entire search space (exploration stage). Lévy flights mimic the peculiar behaviors of living organisms for traveling long distances to deposit pollen over flowers.
- Local pollination considering flower constancy. Under this procedure, the flower constancy among pollination agents and flowers is achieved by local movements. This mechanism enhances the exploitation stage over the search space.
- Switch probability. This procedure controls the switching strategy between global pollination and local pollination during the optimization process.

The MFPA method extends the search capabilities of the original structure of FPA to locate multiple optima at a single execution. Most of the application of evolutionary optimization algorithms tend to converge into only one global optimum. However, the global solution may be composed by expensive or impractical parameters. Under such circumstances, the MFPA efficiently identify multiple global and local optima. The MFPA operates using three computational procedures. The first procedure is called *Memory Mechanism* (*MM*), which involves the use of a memory structure to maintain not only good fitness solutions but also bad fitness solutions as potential optima solutions. The second procedure called *Selection Strategy* (*SS*), consists of selecting individuals directly from the memory structure to increase the population diversity of the solutions. The third procedure called *Depuration Procedure* (*DP*) is a depuration mechanism to avoid redundant solutions. To operate over these procedures, MFPA divides the evolutionary process into three stages. The first stage ($s = 1$) involves from 0 to 50% of the maximum number of iterations (*MAX_ITER*). The second stage ($s = 2$) involves from 51 to 90%. And the third state ($s = 3$) involves from 91 to 100%.

Under this scheme, MFPA obtains a collection of solutions which correspond to optimal values for a given objective function. From an optimization point of view, the objective function to be solved by the MFPA (considering a minimization approach) can be generalized as follows:

$$\mathbf{X}^{best} = \arg\min J\left(\mathbf{X}\right) \tag{1}$$

where $\mathbf{X}^{best}$ corresponds to the collection of solutions which minimizes the objective function J during the optimization process. $\mathbf{X}$ corresponds to the population of individuals. In MFPA, a population of individuals at k-th iteration can be represented as $\mathbf{X}^k = \left\{\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_n^k\right\}$, where n corresponds to the total number of individuals within the population. Additionally, each element of the population represents a d-dimensional solution vector: $\mathbf{x}_i^k = \left\{x_{i,1}^k, x_{i,2}^k, \ldots, x_{i,d}^k\right\}$, where $i \in \{1, 2, \ldots, n\}$ indexes a certain solution within the n-th element within the population. In the following subsections, the multimodal optimization process of the MFPA is described in terms of the three computational procedures.

A. MEMORY MECHANISM (MM)

In multimodal optimization problems, global and local optima must be correctly identified. To indicate whether a solution can be potential optimum, it must fulfil with two characteristics; it represents good fitness values and it is considered as the best solution within its neighborhood. Under such circumstances, the *MM* mechanism of the MFPA uses a memory structure to register potential candidate optima considering the previously characteristics. This memory structure also considers not only new solutions but also past solutions during the evolutionary process. The memory structure is composed by an array of g-th $\mathbf{M} = \left\{\mathbf{m}_1, \mathbf{m}_2, \ldots \mathbf{m}_g\right\}$ elements, where each element corresponds to a d-dimensional solution vector $\left\{m_{w,1}, m_{w,2}, \ldots, m_{w,d}\right\}$ in the optimization problem, where $w \in \{1, 2, \ldots, g\}$. To achieve a correctly memory registration for a given element, the *MM* procedure is conducted over two stages: memory initialization and memory capture. The memory initialization stage only occurs when $k = 0$. In this stage, the best solution in the population is registered into the memory. Once this initialization procedure is complete, the capture phase is conducted according to the following description.

1) CAPTURE PHASE

The main objective of this stage is to register potential optima during each iteration of the evolutionary process. To register a solution $\mathbf{x}_i^k$ into the memory structure, $\mathbf{x}_i^k$ is tested applying two fitness value rules. These fitness rules fulfil the two previously described features to register potential optima into the memory structure.

a: SIGNIFICANT FITNESS VALUE RULE

Under this rule, the comparison of the fitness value for the solution $\mathbf{x}_i^k$ and the fitness value of the worst element inside the memory $\mathbf{m}^{worst}$ is conducted. This rule considers the solution $\mathbf{x}_i^k$ as a candidate global or local optimum if its fitness value is better than the fitness value of $\mathbf{m}^{worst}$. After this, the rule considers whether the solution $\mathbf{x}_i^k$ corresponds to a new optimum or it is like other element inside the memory. To compute this similarity, the normalized distance $\|D_{i,g}\|$ among $\mathbf{x}_i^k$ solution respect to the rest of elements inside the memory structure $\mathbf{M}$ is calculated as follows:

$$\|D_{i,g}\| = \left\| \left(\frac{\mathbf{x}_{i,1}^k - \mathbf{m}_{g,1}}{l_1^{upper} - l_1^{lower}} \right)^2 + \left(\frac{\mathbf{x}_{i,2}^k - \mathbf{m}_{g,2}}{l_2^{upper} - l_2^{lower}} \right)^2 + \ldots + \left(\frac{\mathbf{x}_{i,d}^k - \mathbf{m}_{g,d}}{l_d^{upper} - l_d^{lower}} \right)^2 \right\| \tag{2}$$

where $l_{\{1,2,\ldots,d\}}^{upper}$ and $l_{\{1,2,\ldots,d\}}^{lower}$ corresponds to the fixed upper and lower bounds within the search space defined by (1). And $\mathbf{m}_g$, represents an element of the memory. Next, it is computed a probability function parameterized by the current state of the evolutionary process s as follows:

$$P\left(\|D_{i,n}\|, s\right) = \|D_{i,n}\|^s \tag{3}$$

where $\|D_{i,n}\|$ corresponds to the distance among $\mathbf{x}_i^k$ and the nearest element of memory $\mathbf{m}_n$. The effect of the current state, s (3), indicates that large distances will be promoted at early iterations of the evolutionary process increasing the probability that the solution $\mathbf{x}_i^k$ belongs to the memory structure. In counterpart, in final iterations of the evolutionary process, small distances will be computed decreasing the probability that the solution $\mathbf{x}_i^k$ belongs to the memory structure. Finally, to decide if the solution represents a new optimum or similar solution respect other elements inside the memory structure, a random number α is produced. Then, if α is less than $P\left(\|D_{i,n}\|, s\right)$, the solution $\mathbf{x}_i^k$ will be considered as a new optimum hence, it will be registered into the memory. Otherwise, the fitness value of the solution $\mathbf{x}_i^k$ is compared to the fitness value of the $\mathbf{m}_n$ memory element. The complete significant fitness value rule can be generalized by the following:

$$\mathbf{M} = \begin{cases} \mathbf{m}_{g+1} = \mathbf{x}_i^k, & \alpha < P\left(\|D_{i,n}\|, s\right) \\ \mathbf{m}_n = \mathbf{x}_i^k, & \text{if } J\left(\mathbf{x}_i^k\right) < J\left(\mathbf{m}_n\right) \end{cases} \tag{4}$$

b: NON-SIGNIFICANT FITNESS VALUE RULE

The Significant Value Rule involves the incorporation of a given solution into the memory structure if the solution represents good fitness value. Contrary to this, the Non-Significant Fitness Value Rule will register solutions with worse fitness values. Under this rule, the incorporation of the solution $\mathbf{x}_i^k$ is based on the computation of the probability of the relationship among the $\mathbf{x}_i^k$ and the best $\mathbf{x}^{best}$ and worst $\mathbf{x}^{worst}$ solutions currently available in the population. The probability function is then computed as follows:

$$p = 1 - \frac{J\left(\mathbf{x}_i^k\right) - J\left(\mathbf{x}^{best,k}\right)}{J\left(\mathbf{x}^{worst,k}\right) - J\left(\mathbf{x}^{best,k}\right)} \tag{5}$$

Then, (6) is calculated. In this equation, the probability P corresponds to a probability greater than 0.5 if the solution $\mathbf{x}_i^k$ have good fitness value.

$$P = \begin{cases} p, & 0.5 \leq p \leq 1 \\ 0, & 0 \leq p < 0.5 \end{cases} \tag{6}$$

Finally, to test whether the solution $\mathbf{x}_i^k$ can be part of the memory structure, a random value β must be less than the P probability from (6). Then, the normalized distance from (2) is computed to obtain the shortest distance value determined by the n-th element of the memory ($\mathbf{m}_n$). For the inclusion of the solution $\mathbf{x}_i^k$ into the memory structure, the following rule is applied:

$$\mathbf{M} = \begin{cases} \mathbf{m}_{g+1} = \mathbf{x}_i^k, & \text{if } \gamma < \|D_{i,n}\|^s \\ \text{no change}, & \text{Otherwise} \end{cases} \tag{7}$$

where γ is a random number among the range [0-1], which indicates if the tested solution must be part as a member of the memory structure or not. This scheme contributes to increase the population diversity within the memory structure. s corresponds to the actual state of the evolutionary process.

B. SELECTION STRATEGY (SS)

This procedure extends the capabilities of the original elitist selection criterion from the FPA to incorporate multimodal capabilities. MFPA considers a selection mechanism for capturing potential global and local optima. In the *SS* strategy, a new population $\mathbf{X}^{new}$ is composed by the first n memory elements, where n corresponds to the size of the population. The selection mechanism replaces each individual of the original population for each element belonging to the memory structure to improve population diversity over the entire optimization process.

C. DEPURATION PROCEDURE (DP)

The *MM* and *SS* mechanisms of MFPA, allows multiple optima to be registered into the memory structure at each iteration of the optimization process. However, the memory can contain element redundancy for that, a depuration procedure must eliminate element redundancies insider the memory structure improving the identification of valid and significant optima. The execution of the *DP* mechanism occurs at the end of each stage s of the evolutionary process.

In MFPA, the *DP* procedure computes the distances between memory elements to eliminate concentrations of redundant elements while maintains the most significant memory elements. To accomplish this task, it is obtained the distance among elements within the memory. Then, a depuration ratio is computed for each element of the memory considering the midpoint between each pair of memory elements. Then if the fitness value of the midpoint among two memory elements $J\left(\left(\mathbf{m}^i + \mathbf{m}_r\right)/2\right)$ is not worse than $J\left(\mathbf{m}^i\right)$ and $J\left(\mathbf{m}_r\right)$, $\mathbf{m}_r$ is cataloged as part of the same neighborhood (concentration) of the solution $\mathbf{m}^i$. Otherwise, the $\mathbf{m}_r$ element is considered part of another concentration. If the last condition occurs, then the 85% of the distance among the solution $\mathbf{m}^i$ and the $\mathbf{m}_r$ element is considered as a maximum depuration ratio. Finally, all the memory elements inside the neighborhood generated by the depuration ratio are removed.

III. PROBLEM FORMULATION

In this paper, the multi-circle detection has been formulated as a minimization problem which objective is to obtain global and local solutions. Where each solution represents a circle in a given image. The mathematical description of the multi-circle detection problem is defined as:

$$J\left(\mathbf{C}\right)=-\frac{\sum_{h=1}^{N_s}E(c_i)}{N_s} \tag{8}$$

where **C**, corresponds to any circumference (global or local optima) which minimizes the correspondence function $E(\cdot)$ regarding the number of pixels N_s of the i-th circumference (c_i). The correspondence function $E(\cdot)$ is defined as:

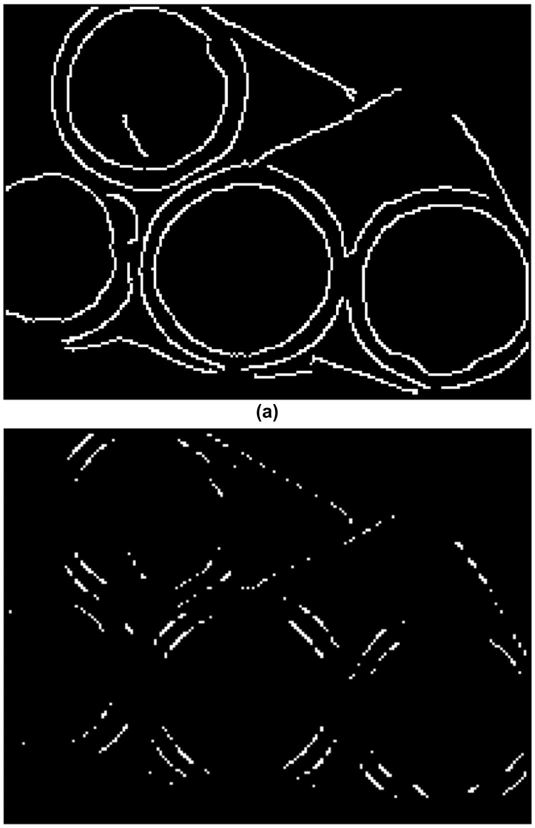
$$E\left(c_i\right)=\begin{cases}1, & \text{if the pixel } (x_i, y_i) \text{ is an edge point}\\ 0, & \text{Otherwise}\end{cases} \tag{9}$$

where (x_i, y_i) corresponds to a pixel belonging to a edge point of the circumference c_i.

IV. MFPA FOR MULTI-CIRCLE DETECTION

In counterpart to unimodal optimization problems, multimodal optimization problems yield to a set of possible global and local optima solutions. Under such circumstance, the set of optimal solutions depends directly on the context of the application. In multi-circle detection problem, it is required to obtain most of the circles within an image in a single run. In this context, the computational procedure for multi-circle detection problem operates over two different domains in order to implement the correspondence function of (8). The index space domain involves an edge map obtained by a pre-processing stage. Then, each solution of the MFPA is transformed into circle space domain to compute an estimated circumference.

As a result, the implementation of the MFPA involves several stages to minimize (8). At first, a pre-processing stage must be applied to eliminate noise and false positive circles from source image. In the pre-processing procedure, a Gaussian filter has been applied to each benchmark image. To obtain an edge map from the original image, the Canny edge detector has been applied. After the Canny algorithm, four types of dilatation operations have been applied to the edge map with the objective of reduce the search space i.e., the edge pixels in the edge map. These dilatations remove lines that are 3 pixels length and with angles of 45°, 90°, 135° and 180° (measured in a counterclockwise direction from the horizontal axis). The effect of such dilatations can be seen in Fig. 1. In comparison to other pre-processing filters, the Gaussian and Canny filters present several advantages such noise reduction, edge preservation and smoothness. The Gaussian filter increase the structural information of the edges rather than other filters where structural edge information may be lost. On the other hand, the Canny filter is more sensible to compute significant edge information. Since the proposed approach operates over an edge map, the required images for the optimization process, must contain robust edge information. For this paper, the standard deviation for both Gaussian and Canny filters was configured by 2. And for the Canny filter, the lower threshold was set by 0.1 and 0.3 the highest threshold, respectively.


(a)

(b)

FIGURE 1. **Image dilatation for edge map. (a) Edge map without dilatations. (b) Edge map with dilatations.**

From Fig. 1, it can be shown that the image dilatations can remove noisy patterns in the edge map and preserve enough circle pixels to detect them.

After the pre-processing stage is achieved, the initialization process of the MFPA considers a population of individuals (circles) in the index space domain. Then, each solution is transformed into a parametric representation of a circle to obtain an estimated circumference to compute the correspondence among the estimated circumference and the edge map. Also, the multimodal capabilities of the MFPA register potential circles in a memory structure. Finally, a depuration procedure eliminates redundant circles obtain by the exploration and exploitation operators of the MFPA. The implementation of the computational procedures for MFPA to solve multi-circle detection problems is described in the following sub-sections.

A. INITIALIZATION

In MFPA (as well as the rest of tested evolutionary methods), an individual (circle) c_i is composed by a *3*-dimensional vector $c_i = (p_i, p_j, p_k)$ where p_i, p_j and p_k corresponds to three randomly chosen points of the edge map. In this representation, each solution is expressed in the index space domain. To compute the fitness value (8), a circle space domain representation is required for each solution to obtain the estimated circumference. In this representation, a solution is encoded by a *3*-dimensional vector $c_i = (x_0, y_0, r)$ where x_0 and y_0 corresponds to the circle's center and r represents the circle's radius. The transformation of index space domain and circle space domain is based on the second-degree equation of a circle as follows:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \tag{10}$$

To compute the *3*-dimensional vector $c_i = (x_0, y_0, r)$, the following mathematical description is defined as:

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} x_j^2 + y_j^2 - \left(x_j^2 + y_i^2\right) & 2 \cdot \left(y_j - y_i\right) \\ x_k^2 + y_k^2 - \left(x_i^2 + y_i^2\right) & 2 \cdot (y_k - y_i) \end{bmatrix} \mathbf{B} \\
&= \begin{bmatrix} 2 \cdot \left(x_j - x_i\right) & x_j^2 + y_j^2 - \left(x_i^2 + y_i^2\right) \\ 2 \cdot (x_k - x_i) & x_k^2 + y_k^2 - \left(x_i^2 + y_i^2\right) \end{bmatrix}, \\
x_0 &= \frac{\det(\mathbf{A})}{4\left(\left(x_j - x_i\right)(y_k - y_i) - (x_k - x_i)\left(y_j - y_i\right)\right)}, \\
y_0 &= \frac{\det(\mathbf{B})}{4\left(\left(x_j - x_i\right)(y_k - y_i) - (x_k - x_i)\left(y_j - y_i\right)\right)}, \\
r &= \sqrt{(x_0 - x_d)^2 + (y_0 - y_d)^2}, \quad d \in \{i, j, k\}
\end{aligned} \tag{11}$$

The mathematical description in (11) corresponds to the computation of the circle center (x_0, y_0, and the radio r based on the circle equation [28]. A graphical description for the index space domain and the circle space domain can be shown in Fig. 2.

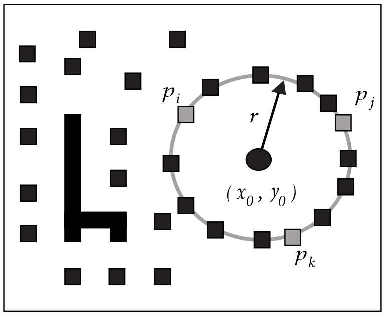


FIGURE 2. Candidate circle (solution) c_i from index space domain and circle space domain.

Then, the approximated circumference obtained by the circle space representation is computed using the Bresenham's Algorithm [29].

B. MEMORY MECHANISM (MM)

In the capture phase of the *MM* mechanism, each solution is compared against the worst element within the memory structure to determinate whether the Significant Value Rule will be applied. If that is the case, then the probability $P\left(\left\|D_{i,n}\right\|, s\right)$ described in (3) is calculated; however, the normalized distance $\left\|D_{i,g}\right\|$ is calculated using the circle space representation of the solutions and memory elements, giving as result the (12):

$$\left\|D_{i,g}\right\| = \left\| \left(\frac{\mathbf{x}_{0,i}^k - \mathbf{m}_{g,x_0}}{l_{x_0}^{upper} - l_{x_0}^{lower}}\right)^2 + \left(\frac{\mathbf{y}_{0,i}^k - \mathbf{m}_{g,y_0}}{l_{y_0}^{upper} - l_{y_0}^{lower}}\right)^2 + \left(\frac{\mathbf{r}_i^k - \mathbf{m}_{g,r}}{l_r^{upper} - l_r^{lower}}\right)^2 \right\| \tag{12}$$

where $\mathbf{x}_{0,i}^k$, $\mathbf{y}_{0,i}^k$ and $\mathbf{r}_i^k$ are the coordinates x, y and radius of the circle represented by the i-th solution in the current k iteration; $\mathbf{m}_{g,x_0}$, $\mathbf{m}_{g,y_0}$ and $\mathbf{m}_{g,r}$ are the coordinates x, y and radius of the circle represented by the g-th memory element; $l_{x_0}^{upper}$, $l_{y_0}^{upper}$ and l_r^{upper} are the upper bounds for the coordinates x, y and radius of the circles and $l_{x_0}^{lower}$, $l_{y_0}^{lower}$ and l_r^{lower} are the lower bounds of the circles. After $P\left(\left\|D_{i,n}\right\|, s\right)$ is calculated, the Significant Value Rule behaves exactly as it was defined in (4). In the case where the Significant Value Rule is not applied, the probability P defined in (6) is calculated to determinate whether the Non- Significant Value Rule will be applied. If this rule is applied, then $\left\|D_{i,g}\right\|$ is calculated as described in the (12) and the process continues as described by (7).

C. SELECTION STRATEGY (SS)

The selection strategy is applied at the end of each iteration. Here, the memory size h is compared with the current population size n. If $h > n$, the new population $\mathbf{X}^{new}$ is formed by the best n memory elements; otherwise, $\mathbf{X}^{new}$ is formed by substituting the first h population solutions with all the memory elements.

D. DEPURATION PROCESS (DP)

At the beginning of the depuration process, the circle space representation of the memory elements is calculated. Next, the Euclidean distance and midpoint $p_{g,i}$ between each one of the circle space representation of the memory elements m_g and m_i is calculated, being $g \neq i$. Then, the midpoint $p_{g,i}$ is evaluated in the objective function; if the fitness value of $p_{g,i}$ is not worse than the fitness values of m_g and m_i, m_i is considered being part of the same concentration of solutions around m_g. However, if the fitness value of $p_{g,i}$ is worse than the fitness values of m_g and m_i, the element m_i is considered part of another concentration of solutions and thus, the 85% of its distance with m_g will be saved as a depuration range. Finally, all the solutions inside the depuration range of each one of the m_g solutions are removed from the memory.

V. EXPERIMENTAL RESULTS

In this section, the performance of the MFPA has been compared against three commonly used unimodal evolutionary approaches; namely Differential Evolution (DE) [30], Harmony Search (HS) [32] and Grey Wolf Optimizer (GWO) [15]. Additionally, the performance of the proposed method is compared against Multimodal Particle Swarm Optimization (MPSO) [26]. All the evolutionary algorithms used in this experimental study, have been proposed to solve multi-circle detection problem. DE [31], HS [33], and GWO [34] by a unimodal fashion, and MPSO considering a multimodal perspective [35].

Due to the lack of standard benchmark dataset for circle detection problem, a dataset conformed by synthetic and real images is considered. The proposed dataset in this paper consists of 25 images. 15 images are synthetic, and 10 images are real-world photos where circular shapes are contained. For all the images considered in the dataset, the resolution was 640 × 480 pixels. The synthetic images incorporate several complexities regarding different geometric shapes and noise. The experiments have been performed using MATLAB R2022b running over Windows 11 Pro, 12th Gen Intel Core i7-CPU, 3.60 GHz with 16 GB RAM. Furthermore, for each image a list of Ground Truth Circles (GTC) $G^I = \{(x_1, y_1, r_1), (x_2, y_2, r_2), \ldots, (x_m, y_m, r_m)\}$ is also defined, where m is the total number of GTC on an image I. The GTC list is used to measure the precision of circle location, size and number of circles found by the tested evolutionary algorithms. For the performance comparison, several indexes have been considered in the experimental study. The first performance index is the Vector of Coincidence (VC), which stores information regarding how many, and which circles were found. The coincidences value indicates the number of circles found in the image including redundant solutions. The Effective Circle Number (ECN) computes how many circles each evolutionary approach is able to recognize eliminating redundant circles in the image. Also, the Nearest Circle Distance (NCD) is the distance among an approximated circumference and its corresponding GTC. To avoid random effect, the mean and standard deviations are also calculated to obtain robustness measurements.

To make a fair comparison of the tested algorithms and the experimental study, the number of iterations and the number of solutions in the population has been set to 100 and 200, respectively for each evolutionary scheme. To eliminate the random effect, each algorithm has been executed 30 times for each image. At the end of each single run, only the solutions with a fitness value equal or less than -0.9 were considered to avoid false positive solutions. Additionally, as the DE, HS, and GWO methods do not have multimodal mechanisms to preserve multiple optima through the iterations, redundant solutions have been removed. To maintain compatibility in published literature, each evolutionary method was implemented following their original guidelines proposed by the authors. For the numerical performance comparison, the parameters of each evolutionary methodology, was configured according to published literature related to solve multi-circle detection problem. Such parameter configuration is exposed in Table 1.

TABLE 1. Parameter configuration.

Algorithm	Parameters
DE/rand/1/bin [30]	Crossover probability *cr = 0.8.* Differential weight *dw = 0.25.* According to [31]
HS [32]	*HCMR* = 0.7. *Pitch* = 0.3. According to [33]
GWO [15]	Linearly decreasing update mechanism from the interval [2-0]. According to [34]

MPSO [26]	w = 0.96.
	c1 = c2 =1.2.
	According to [35]

For all the experiments, each particle (circle) considers the low radius threshold was set to $R_{\min} = 10$ pixels, while the high radius threshold value was given by the following equation:

$$R_{\max} = \frac{\min(width_I, height_I)}{2} \tag{13}$$

where $\min(width_I, height_I)$ corresponds to the minimum value in pixels between the width and height of the current image I. Both thresholds were experimentally obtained to eliminate false positives. For the low radius threshold, 10 pixels seemed like a value small enough to do no let the algorithm find little circles formed by the noise in the image, as for the high radius threshold, the value was selected such that the largest possible circle wound not be larger than the image, i.e., it can fit completely in it.

A. PERFORMANCE CRITERIA

To analyze the performance of the MFPA against DE, HS and GWO, a vector of coincidences VC used to measure how many and which circles were found by each evolutionary approach. The number of elements of this vector is equal to the GTC in the analyzed image and each element of the vector corresponds to the number of times that a ground truth circle was found; thus, the first vector element of VC is the number of times that the first ground truth circle was found, the second element is the number of times that the second ground truth circle was found, and so on.

TABLE 2. Original images, edge maps and circles considered as ground truth form the non-synthetic images.

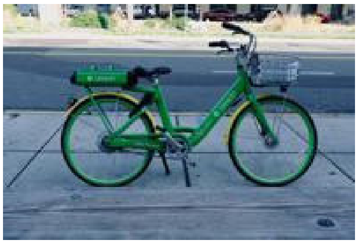
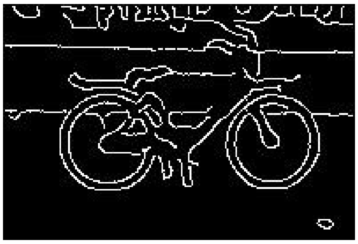
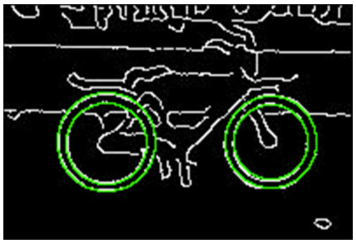
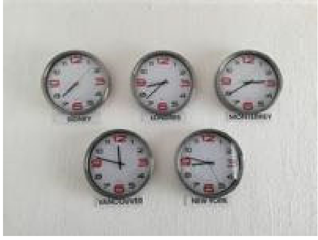
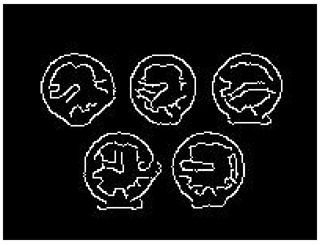
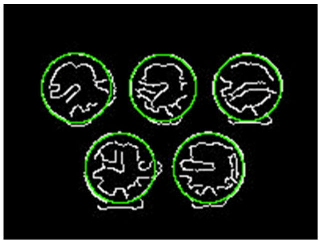
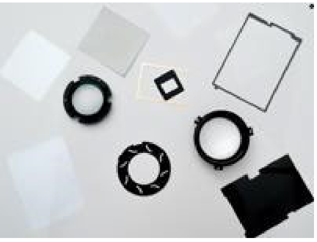
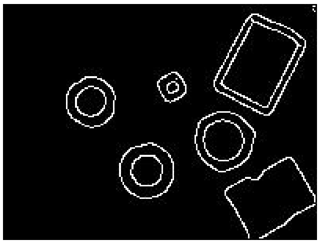
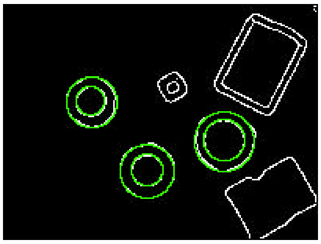
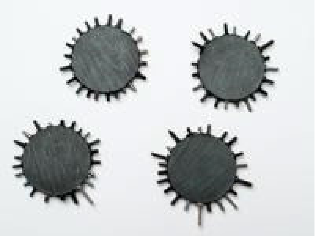
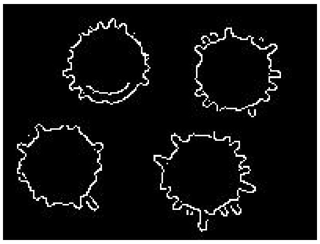
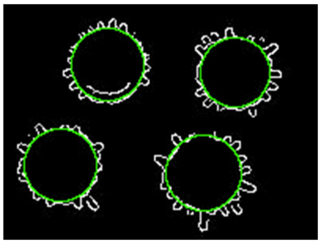
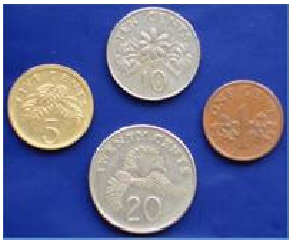
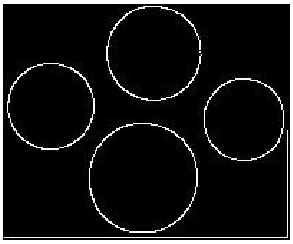
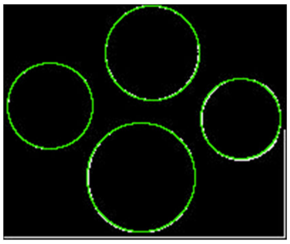
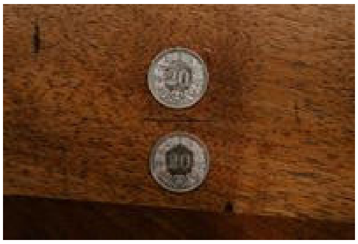
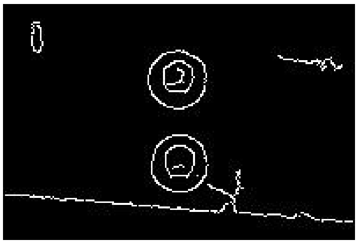
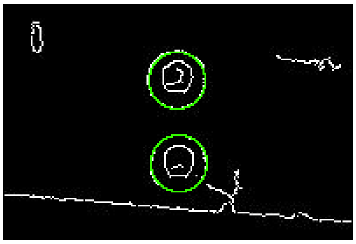
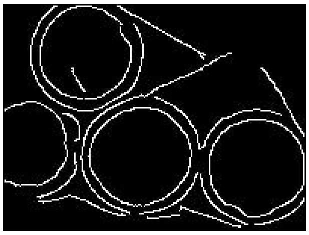
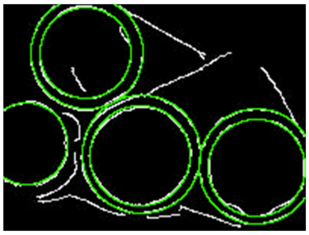
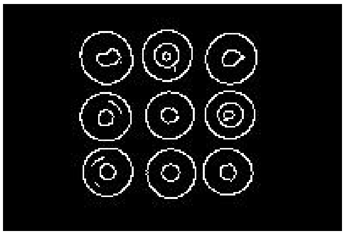
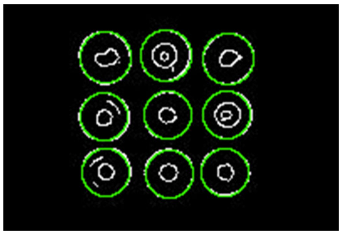
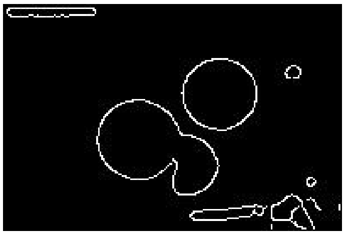
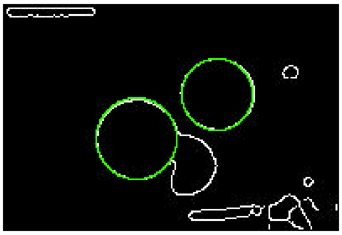
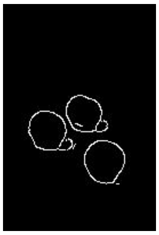
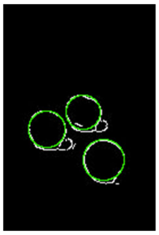
Original image	Edge map	Ground truth
		
		
		
		
		
		

TABLE 2. ***(Continued.)*** **Original images, edge maps and circles considered as ground truth form the non-synthetic images.**

For example, if an image I contains 3 ground truth circles, i.e, $G^I = \{(x_1, y_1, r_1), (x_2, y_2, r_2), (x_3, y_3, r_3)\}$, and the coincidence vector VC contains $VC = (2, 1, 1)$, it means that the algorithm found the circle (x_1, y_1, r_1) 2 times and the circles (x_2, y_2, r_2) and (x_3, y_3, r_3) one time each. Furthermore, in this case it is said that the algorithm obtained 4 coincidences i.e., it found a total of 4 circles.

The coincidence vectors are also used to calculate the Effective Circle Number (*ECN*), which represents the cardinality of the coincidence vector *VC* where each component must be greater than zero. Generally, it can be said that the coincidences metric counts the total number of circles found by the algorithm considering redundant solutions, while *ECN* counts the number of circles found by the algorithm without redundant solutions.

To associate a solution c_i with a ground truth circle GTC_i, the Nearest Circle Distance (*NCD*) between the obtained solution by an algorithm and each of the GTC is calculated as follows:

$$NCD = \frac{|c_i - GTC_i|}{|GTC_i|} \tag{14}$$

where c_i is the i-th solution expressed in the circle space, GTC_i is de i-th ground truth circle and || is the norm function. The *NCD* metric represents the minimum error which measures how close a solution c_i is to the nearest ground truth circle. The *NCD* is used as a measure of the precision of a solution, as the larger this value is, more different the solution is from the ground truth circle to which is more alike.

To eliminate the random effect, the mean $\mu(VC)$ and standard deviation $\sigma(VC)$ of the elements of the coincidence vectors are computed. In the ideal scenario, each algorithm finds all the circles once, resulting in $\mu(VC) = 1$ and $\sigma(VC) = 0$. If $\mu(VC) < 1$ then, the algorithm did not find all the circles in the image, and if $\mu(VC) > 1$, the tested algorithm found more circles than it should. On the other hand, if $\sigma(VC) > 0$ then, the algorithm found some circles more times than others.

It is important to notice that $\mu(VC)$ and $\sigma(VC)$ are metrics that complement each other, as the mean could be equal to 1 even if not all the circles were found, or the standard deviation could be 0 even if the number of circles found is not correct. So, to have a good insight of the performance of the algorithm both values should be analyzed. For example, in the case that $VC = \{0, 1, 0, 3\}$, despite that $\mu(VC) = 1$, 2 circles were not found, and one was found 3 times. This scenario is reflected in the value of $\sigma(VC)$ which in this case is 1.4142. Another example is the case $VC = \{3, 1, 2, 1\}$; here $\sigma(VC) = 0$, but the algorithm found some circles multiple times, which is shown by the value of $\mu(VC)$ that in this case is 1.75.

B. PERFORMANCE COMPARISON

In this section, the numerical results for the performance comparison among MFPA and the rest of tested algorithms is presented. To measure the performance of each evolutionary algorithm, the performance indexes from Section V-A must be computed. Since the performance metrics evaluate the GTC, a definition of GTC is required.

For the synthetic images, the GTC can be easily identified as these only consist in simple black shapes on a white background. However, this is not the case in some of the real images due to observer interpretation. To avoid ambiguities in real images, the circles considered for the experimental study had been selected by analyzing the edge maps of the images, as the shapes can easily be identified. The edge maps and the ground truths that were used in the experiments can be shown in Table 2.

For the performance analysis, Tables 7, 8 and 9 (These tables are organized in Appendix A) report the numerical results for each algorithm. In the tables, it is exposed each GTC considering the form (x, y, r). Table 7 contains 10 synthetic images with different complexities regarding different geometric shapes. Table 8 contains 10 real photos containing circle shapes. Finally, Table 9 exhibit the performance results of synthetic images containing noise to test the scalability of the proposed method and the rest of tested algorithms. Each experiment has been conducted through 30 independent runs to avoid random effect. According to the *ECN* metric, Table 7 suggests that only the MFPA method obtains the total number of circles for each image. These results indicate that the proposed method is able to capture all potential optima within a multimodal environment. Under such situation, MFPA obtains each circle without redundancy. This is corroborated by the *VC* metric. Since the *VC* vector measures the coincidences among GTC and approximated circles, it is considered only the cardinality of the *VC* vector. Contrary to the MFPA, the rest of evolutionary methodologies considered in the experimental study do not detect efficiently most of the circles. According to the *NCD* metric, which computes the error distance among the GTC and the approximated circles by each image, almost all the tested algorithms produce good results comparing their distances among an approximated circle with a ground truth circle. The numerical results of the MPSO, indicate that the multimodal strategy of MPSO, does not obtain reliable results since it computes multiple false positive outcomes.

From the numerical results from Table 7, the proposed method is the only approach that obtains most of the ground truth circles with consistent results, also considering the results of the MPSO, it is quite clear that the multimodal structure of the MFPA outperforms the multimodal strategy for MPSO in terms of robustness. The memory management of the MFPA efficiently captures potential optima using a multimodal optimization approach for multi-circle detection in a single run. Contrary to the rest of algorithms, these algorithms are not capable to detect multiple optima in a single run.

Table 7, indicates that the proposed method is able to detect most of the circles even when overlapping circles, overlapping shapes and complex geometric shapes are presented. This is achieved by the balance of the exploration and exploitation operators and the memory management rules to produce non-redundant solutions.

The multimodal nature of the multi-circle detection problem is an evident approach of the lack of capabilities of unimodal optimization algorithms to detect multiple circles in a single run. Under such circumstances, the outstanding performance of the proposed multimodal algorithm achieves a better alternative to the multi-circle detection problem.

To test the scalability of the proposed method against unimodal optimization methods, Table 8 exhibits the numerical results of the performance analysis considering real photos containing multiple circular shapes.

According to Table 8, MFPA outperforms the rest of tested algorithms in terms of the *ECN* metric. By analyzing the *ECN* metric, the MFPA method presents some issues even when it performs better than its competitors. This is due to the inherent artifacts of real photos. Commonly, real photos contain several image artifacts such as: color similarity, sharp contrasting edges and color sensibility. However, even when these issues are presented, the proposed approach is able to approximate most of the ground truth circles. As it can be deduced from the numerical results of the *NCD* method, the proposed approach obtains the lowest error distances among GTC and approximated circles than DE, GWO, and HS methodologies. These results are validated considering the *VC* vector which corroborates the *ECN* metric.

It can also be shown that from some images, no algorithm can obtain a single circle, This could be caused by the position of the circle in the image making this optimal harder to be found, also, some parameters could be needed to be set in specific values in order to find it or even the number of circles in the image could be too large for the number of solutions in the populations and so, a larger number of solutions could be needed.

Even with some of these issues, the proposed method outperforms its competitors in most of the cases.

The numerical results obtained by the MPSO indicates that the multimodal methodology is not capable of finding multiple circles efficiently. On the other hand, the MFPA is capable to locate potential circle solutions in a more efficient multimodal approach than unimodal approach for the multi-circle detection problem.

On the other hand, synthetic images containing noise are interesting test images to evaluate the performance of a multi-circle detection problem. Since noisy synthetic images contain point noise, the optimization process can present difficulties since each point noise contributes to the sensibility of a circle detection problem. Under such circumstances, synthetic images with noise can significantly restrict the performance of any optimization technique. Additionally, to the image noise, the obtained edge map to compute the objective function adds image artifacts to the images. The Canny algorithm produces duplicate number of circles in the image. This effect can be observed in Fig. 3.

According to Table 9, only the MFPA is capable to detect most of the circles in a single run even with the presence of noise. The structure of the MFPA method allows to operate over noisy images containing overlapped and duplicate circles. It is quite evident that the rest of unimodal and multimodal methods are prone to fail against image noise.

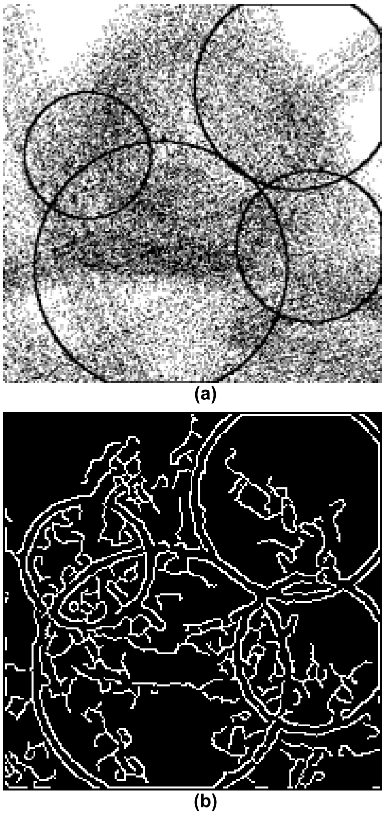


FIGURE 3. Comparison between an image and its edge map. (a) Original image. (b) Edge map obtained by Canny algorithm to detect inner and outer edges of the contours.

In Table 3 are shown the most representatives examples of the outcomes for each evolutionary algorithm for every image in the dataset from Tables 7, 8 and 9. In Table 3, it can be visually demonstrated that almost every image obtained by the MFPA method found every circle, while the other algorithms struggled to find all of them as more complex the images become (for example, images 18, 21, 22, 23, 24 and 25). In images 8 and 9 it is more notorious the problem mentioned previously concerning circles defined by contours, however this exposed another advantage of multimodal approach which is the capacity of finding circles even if they are relatively close to each other.

From the visual results in Table 3, it can be demonstrated that the proposed methodology outperforms the rest of tested algorithms commonly used in the multi-circle detection problem. The balance among exploration and exploitation in combination of the memory structure of the MFPA method produce the best outcomes. The visual results indicate that the MFPA is able to find most of the circles even in complex images containing noise. The multimodal capabilities of the MFPA allows the accurate detection of most of the circles in a single run of the evolutionary method.

C. SENSITIVITY ANALYSIS

In this section, a sensitivity analysis is conducted in order to gain insight about how the multimodal nature of the algorithm affects the experimental outcomes. For that, an example of each type of image (noiseless synthetic images, real photos and noisy synthetic images) is selected to perform new experiments in which the values of $\mu(CV)$ and $\sigma(CV)$ are analyzed when the percentage of the distance between memory elements used in the depuration procedure as depuration ratio is changed. The considered values by the analysis were 5%, 15%, 25%, %35, 45%, 55%, 65%, 75%, 85% and 95%. The algorithm has been executed 30 runs per image. Furthermore, the test images containing the greater number of ground truth circles per image set were chosen. In Tables 4, 5 and 6 are shown the mean values of the outcomes of those 30 runs and the plots of such values can be seen in Figures 4, 5 and 6.

In Table 4 and Fig. 4, it is clear how $\mu(CV)$ and $\sigma\ (CV)$ values start being relatively high and stabilized near the ideal scenario value when the depuration ratio percentage value is greater than 25%. However, $\mu(CV)$ reached its best value at 85% and $\sigma\ (CV)$ at 75% (here, best value means that they are closer to the ideal scenario value). It is also interesting to notice that, when the depuration ratio percentage value is greater than the one in which $\mu(CV)$ and $\sigma\ (CV)$ were closer to the ideal scenario values, $\mu(CV)$ and $\sigma\ (CV)$ start to increase.

In Table 5 and Fig. 5, a similar behavior than the previous case can be seen. Here, both $\mu(CV)$ and $\sigma\ (CV)$ reach their best value when the depuration ratio percentage value is 85% and when the percentage values is 95%, they get far from the ideal scenario value.

The case shown in Table 6 and Fig. 6 is a bit different, here the behavior of $\mu(CV)$ and $\sigma\ (CV)$ as the depuration ratio percentage value increases seems more erratic which could be due the amount of noise in the image. However,

TABLE 3. Most representative examples of the algorithms' outcomes for each image in the dataset.

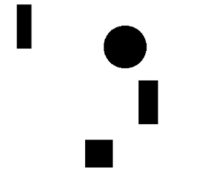
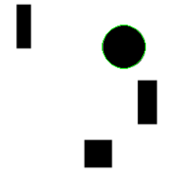
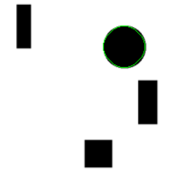
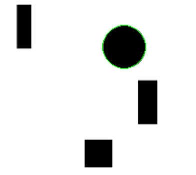
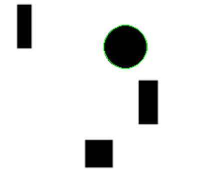
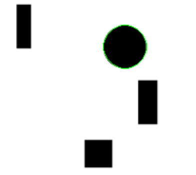
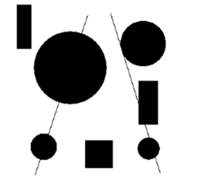
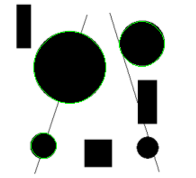
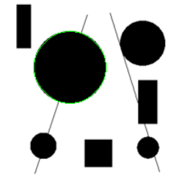
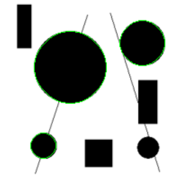
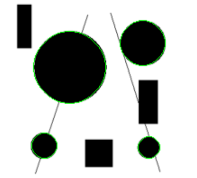
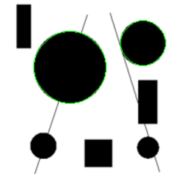
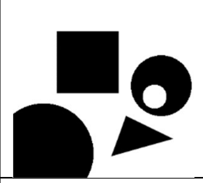
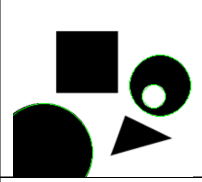
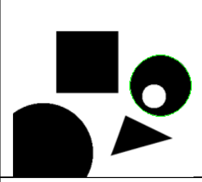
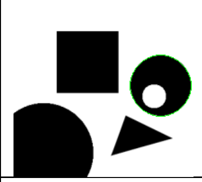
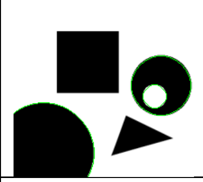
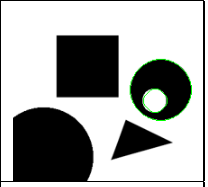
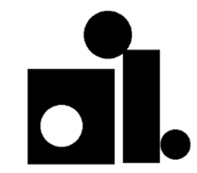
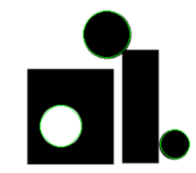
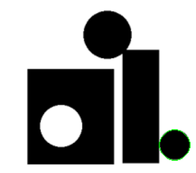
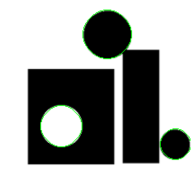
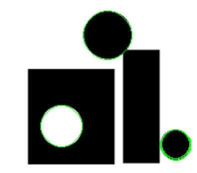
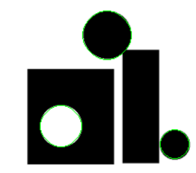
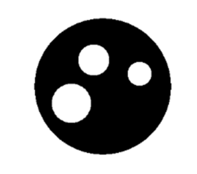
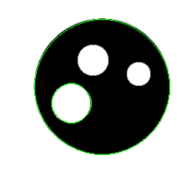
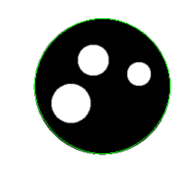
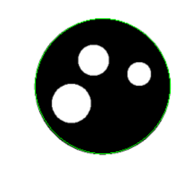
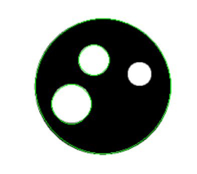
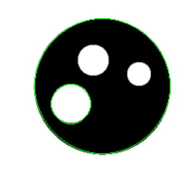
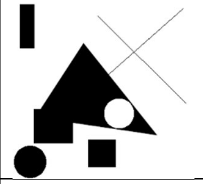
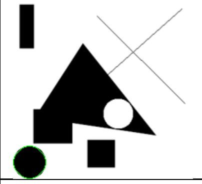
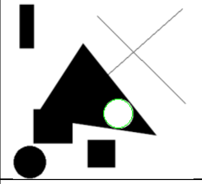
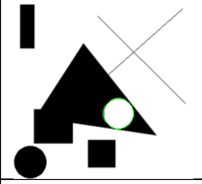
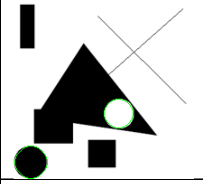
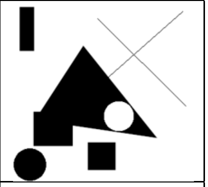
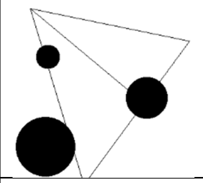
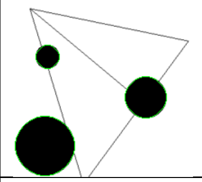
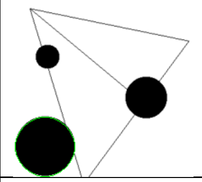
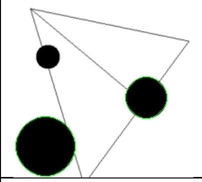
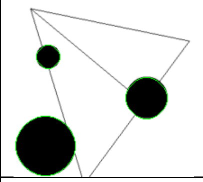
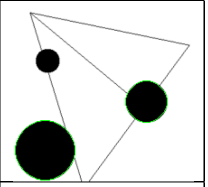
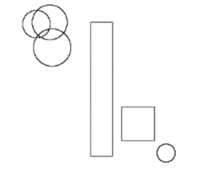
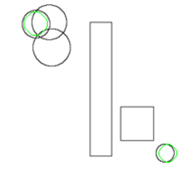
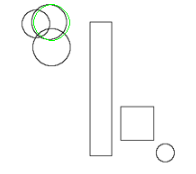
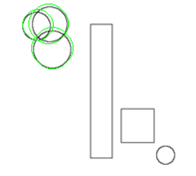
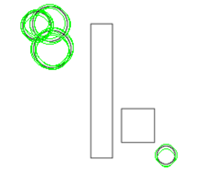
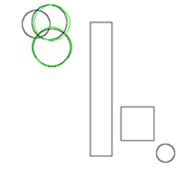
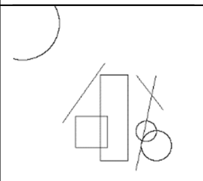
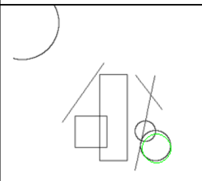
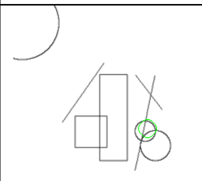
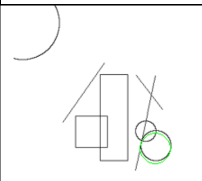
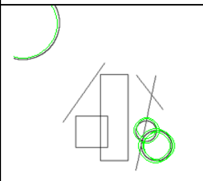
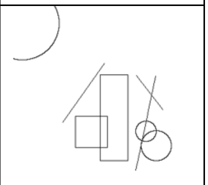
#	Original image	Algorithms				
		DE	GWO	HS	MFPA	MPSO
1						
2						
3						
4						
5						
6						
7						
8						
9						

TABLE 3. ***(Continued.)*** **Most representative examples of the algorithms' outcomes for each image in the dataset.**

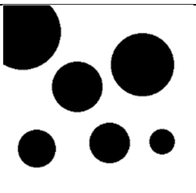
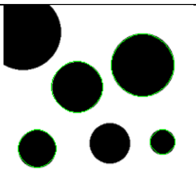
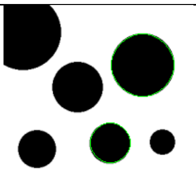
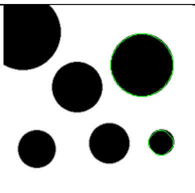
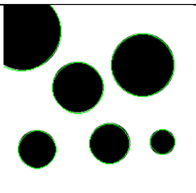
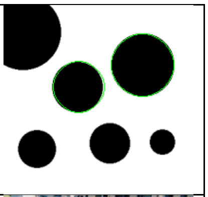
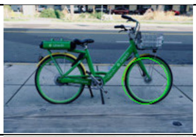
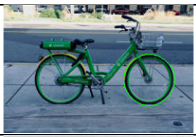
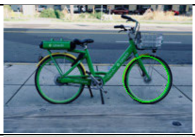
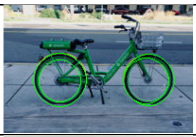
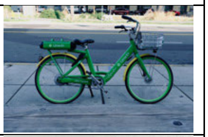
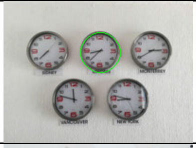
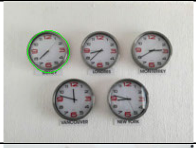
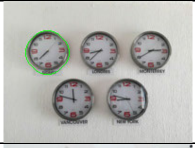
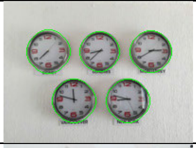
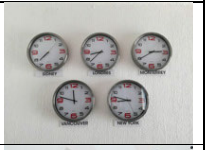
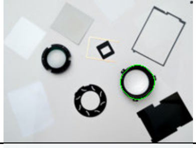
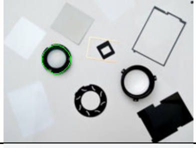
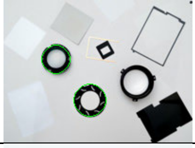
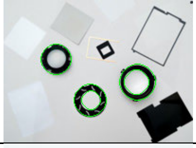
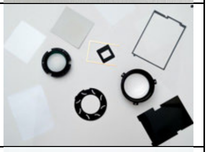
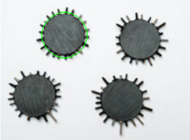
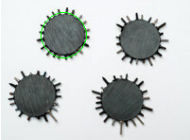
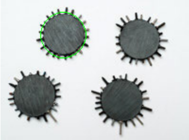
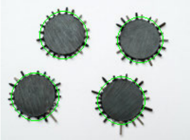
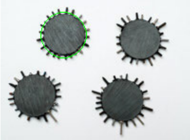
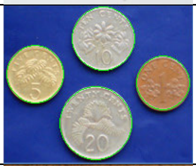
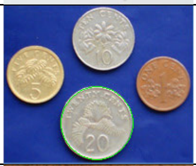
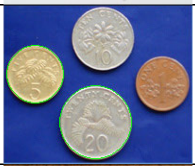
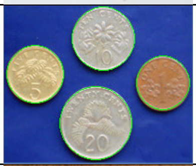
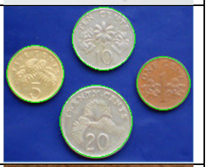
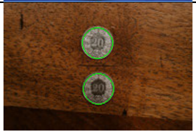
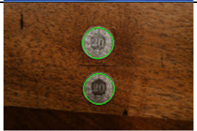
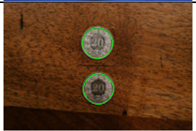
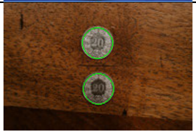
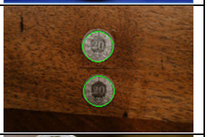
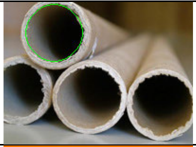
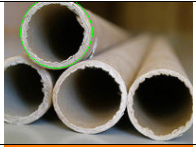
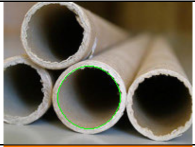
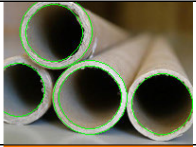
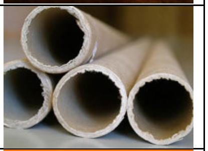
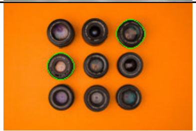
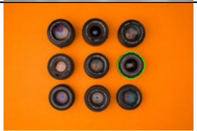
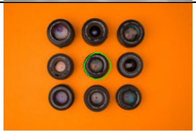
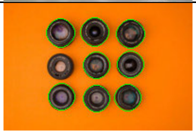
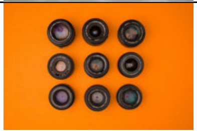
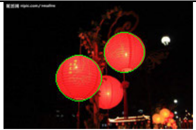
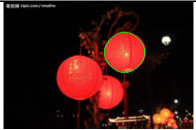
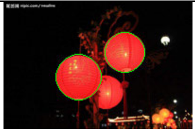
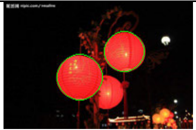
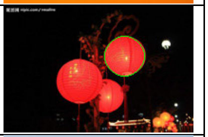
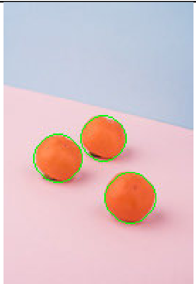
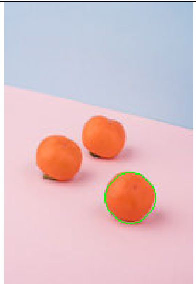
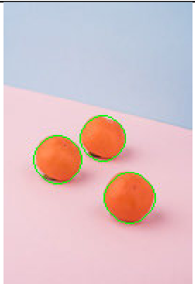
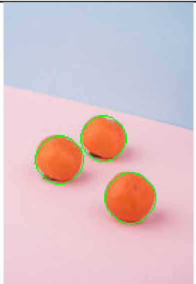
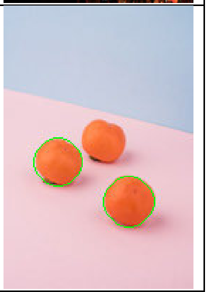
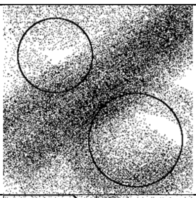
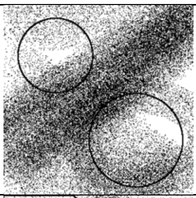
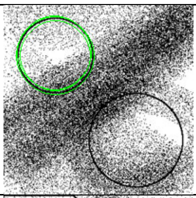
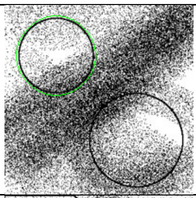
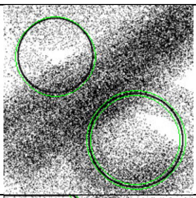
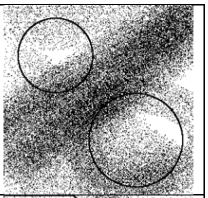
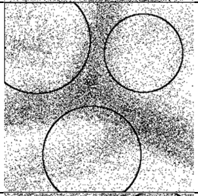
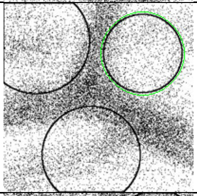
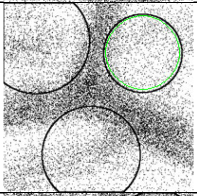
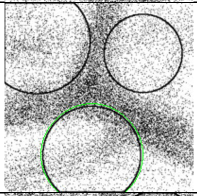
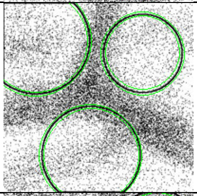
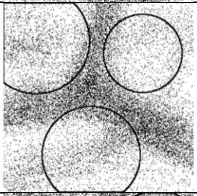
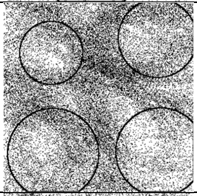
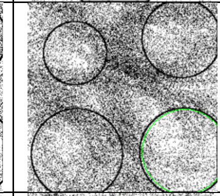
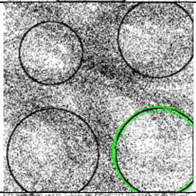
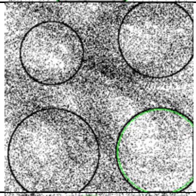
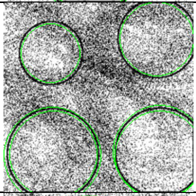
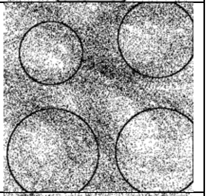
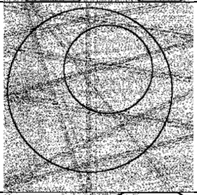
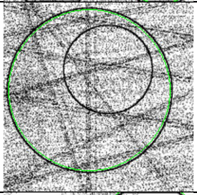
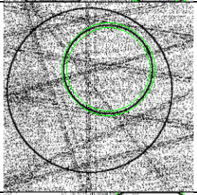
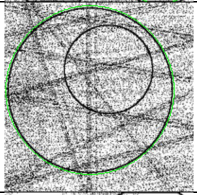
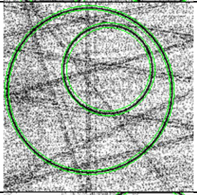
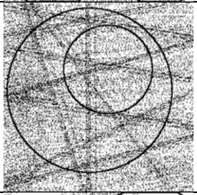
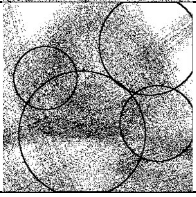
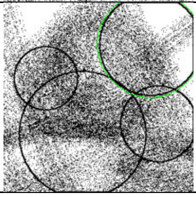
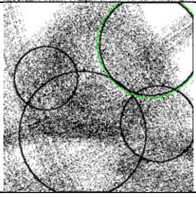
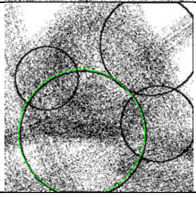
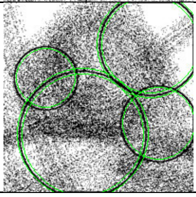
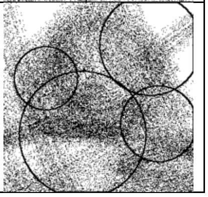
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

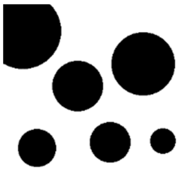
TABLE 3. *(Continued.)* **Most representative examples of the algorithms' outcomes for each image in the dataset.**

21						
22						
23						
24						
25						

it is interesting to notice that, when the depuration ratio percentage value is 85%, $\boldsymbol{\mu}(\boldsymbol{CV})$ and $\boldsymbol{\sigma}$ $(\boldsymbol{CV})$ tend to be near the ideal scenario value. However, different from previous cases, here $\boldsymbol{\sigma}$ $(\boldsymbol{CV})$ gets better value at 95% instead of getting worse.

With this analysis, it can be shown that when the depuration ratio percentage has lower values, the algorithm finds a considerable number of redundant circles in the images. This makes sense because lower depuration ratio percentage values cause the algorithm to identify with grater probability near located solutions in the search space as belonging to different concentrations. Furthermore, it is interesting that in general, $\mu(CV)$ and σ (CV) tend to be closer to the ideal scenario value when the depuration ratio percentage value is greater than 25% but tend to be far from it as the percentage value is higher than 85%. As a conclusion, it can be shown that values near 85% could in general lead to be acceptable results. However, due the erratic behavior described in the third case, this could be different for noisy images and each case should be analyzed independently if results are not the expected.

TABLE 4. μ**(CV) and** σ**(CV) values obtained from running 30 times the algorithm over the image taken from the synthetic noiseless image set.**

Image	Depuration ratio percentage	$\mu(CV)$	$\sigma(CV)$
	5%	1.1333	0.3612
	10%	1.0389	0.1271
	15%	1.0111	0.0755
	25%	1.0111	0.0544
	35%	1.0056	0.0408
	45%	1.0056	0.0408
	55%	1.0000	0.0544
	65%	1.0056	0.0408
	75%	0.9833	0.0680
	85%	0.9889	0.0816
	95%	1.1333	0.3612

D. COMPUTATIONAL EFFORT AND TIME COMPLEXITY

In this section, the computational effort as well as the computational complexity of the evolutionary algorithms is discussed. To compute the computational effort among the tested methodologies, the average execution time for each

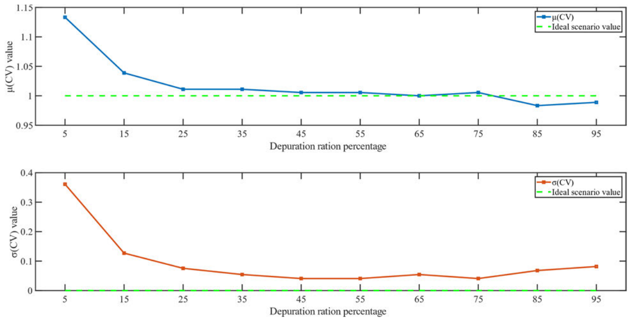


FIGURE 4. Plots of the μ(CV) and σ(CV) values from the values shown in Table 4.

TABLE 5. μ(CV) and σ(CV) values obtained from running 30 times the algorithm over the image taken from the real photo set.

Image	Depuration ratio percentage	$\mu(CV)$	$\sigma(CV)$
	5%	0.8778	0.3338
	10%	0.8778	0.3558
	15%	0.8296	0.3495
	25%	0.8481	0.3201
	35%	0.8444	0.3248
	45%	0.8333	0.3420
	55%	0.8370	0.3260
	65%	0.8556	0.3156
	75%	0.8704	0.2572
	85%	0.8630	0.3009
	95%	0.8778	0.3338

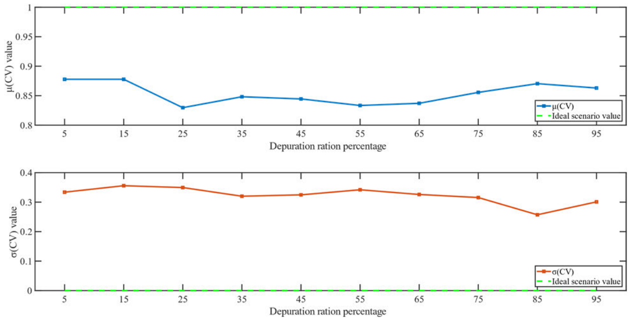
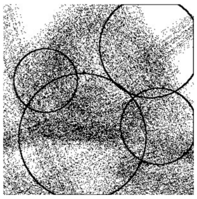


FIGURE 5. Plots of the μ(CV) and σ(CV) values from the values shown in Table 5.

experiment is described in Tables 7-9 in seconds (s). As it can be shown in Tables 7-9, it is quite evident that DE, HS, GWO, and MPSO executes faster than MFPA. However, the obtained results by the MFPA represent more consistent results in terms of the performance indexes. The multimodal mechanism of the proposed MFPA for solving multiple circle detection problem results into a novel perspective which execution time is not so long comparing to the execution time achieved by the GWO. Under such circumstances, the proposed multimodal methodology for circular shape detection is computationally efficient approach compared to unimodal formulation in terms of execution time.

TABLE 6. μ (CV) and σ (CV) values obtained from running 30 times the algorithm over the image taken from the synthetic noisy image set.

Image	Depuration ratio percentage	$\mu(CV)$	$\sigma(CV)$
	5%	2.1417	0.9923
	10%	1.8667	1.0158
	15%	1.9667	1.0033
	25%	1.8750	0.8781
	35%	1.9917	0.8725
	45%	2.0000	0.8519
	55%	1.9167	0.9596
	65%	1.9083	1.0478
	75%	1.9000	0.9041
	85%	1.9750	0.8565
	95%	2.1417	0.9923

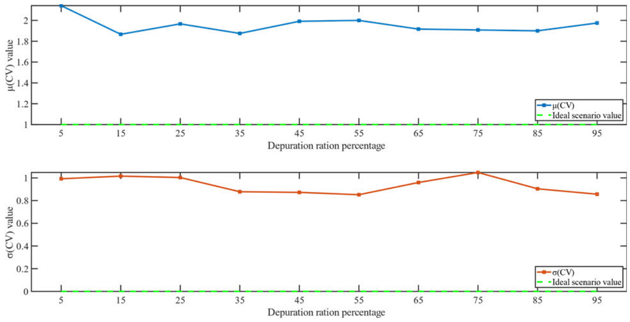


FIGURE 6. Plots of the μ(CV) and σ(CV) values from the values shown in Table 6.

To analyze the computational complexity of the MFPA and the rest of algorithms. It is considered K as the maximum number of iterations, d as the number of considered dimensions for an optimization process, and n as the population size. The computational complexity of the DE variant in the experimental study; DE/rand/1/bin is computed as $O(n \cdot d \cdot K)$ [36]. Also, the computational complexity of the HS algorithm is $O(n \cdot d \cdot K)$ [37]. As it can be shown, either the algorithmic structures of DE and HS are not complex, which makes efficient alternatives to implement to solve multiple optimization problems. On the other hand, the computational complexity of MPSO is $O(n^2)$. However, comparing the performance results in the experimental analysis, it can be demonstrated that GWO and MFPA obtain better results in terms of accuracy and robustness. Under such circumstances, the computational complexity of GWO and MFPA may represent more complex algorithmic structures. The computational complexity of the GWO is divided into two main processes: the leader selection mechanism $O(n)$ and the update position strategy $O(n \cdot d)$ for each iteration ek. For that, the complete complexity is reduced to $O((n + (n \cdot d) \cdot K))$. On the other hand, to analyze the complexity order of the MFPA, the complexity of the original FPA must be analyzed.

The original optimization process of the FPA can be divided into global pollination and local pollinatio. Considering both evolutionary processes in the algorithmic structure, the complexity order is $O(2n)$ which can be reduced into $O(n)$. For that, the complete complexity of the FPA given K iterations is given by $O(n \cdot d \cdot K)$. Finally, since the MFPA considers a multimodal strategy to preserve potential optima in a single run, the complexity order can be divided into the five mechanisms of the MFPA namely, Global pollination, Local pollination, Memory Mechanism, Selection Strategy and Depuration Procedure. Since MFPA uses the Global and Local pollination procedures for the original FPA, the complexity is $O(n)$. For the *MM* mechanism, the complexity is $O(n^2)$ due to the comparison among the entry elements and the stored elements in the memory structure. Also, considering the *SS* procedure, the elitist criterion can consist of preserving only the best solutions during the entire optimization procedure. Under such circumstance, the complexity for the *SS* mechanism is $O(1)$. The last multimodal operator for the MFPA namely, Depuration Procedure, consist of evaluating the Euclidean distance of each memory element to eliminate redundant solutions. Under this perspective, the complexity is given by $O(n^2)$. The entire computational complexity of the MFPA can be computed as $O(n + n^2 + 1 + n^2)$, which can be reduced into $O(2n^2 + n + 1)$. From the previous complexity analysis, it can be deduced that the unimodal evolutionary algorithms presented in this paper achieve linearly time complexity. However, the purpose of this paper is to reformulate the circular shape detection problem into a multiple circular detection. For that, the presented paper incorporates a multimodal version of the original FPA (MFPA), to efficiently store and manage multiple optima, where each solution corresponds to unique circular shape. Under such approach, the MFPA incorporates several procedures to extend multimodal capabilities. Most of multimodal strategies in published literature are based on niching and speciation which incorporates quadratic terms in their algorithmic structures. Under such circumstances, the MFPA also contains quadratic complexity. For that, MFPA is an efficient methodology considering a multimodal optimization approach.

E. DISCUSSIONS AND LIMITATIONS

Most of the published research related to evolutionary algorithms present the design and application of unimodal evolutionary algorithms to solve multimodal optimization problems. Real-world optimization problems commonly yield into unimodal and multimodal mathematical formulations. Unimodal optimization problems relate to the process of finding a single global optimum over objective functions which mostly represent multimodal topologies. However, from a practical point of view, global optima may not always be realizable due to several realistic constraints. For that, unimodal evolutionary algorithms may not be suitable to operate into multimodal topologies. Under such circumstances, multimodal evolutionary algorithms can be an alternative methodology to obtain multiple optimal solutions during the optimization process. In this paper,

multiple circle detection problem within images is reformulated as a multimodal optimization problem. For that, a multimodal evolutionary algorithm which can find multiple circles in a single run is required. The MFPA extends the capabilities of the original FPA to incorporate multimodal functionality in its evolutionary strategy. The multimodal capabilities of the MFPA are divided into three main procedures. The first computational procedure is the Memory Mechanism (*MM*). In this procedure, each solution competes to be preserved in future generations. This mechanism only stores potential optima at each generation in a memory structure. This process is based on the distance and the fitness value of each solution. As a result, the *MM* allows to increase the population diversity at each iteration. The second procedure is the Selection Strategy (*SS*). Under this process, new solutions are added into the memory structure at each iteration. This procedure improves the exploration stage (global pollination operator of the original FPA) of the optimization process by exploring wider zones in the search space in future generations. Since the *MM* and SS mechanisms store new elements into the memory structure, a cleaning process to maintain diversity and balance among exploration and exploitation stages is required. Under such circumstances, the third computational component called Depuration Procedure (*DP*) cleans the memory structure in order to eliminate redundant solutions in the memory avoiding the formation of solutions clusters. This mechanism improves the exploitation stage (local pollination operator of the original FPA) by exploring narrower zones in the search space in future generations. According to the performance results of the MFPA and the rest of tested evolutionary methods, it is quite evident that the combination of the *MM*, *SS* and *DP* procedures in conjunction to the original FPA operators, result into a powerful multimodal scheme for solving multiple circle detection problem. The remarkable performance of the MFPA against the rest of evolutionary methods is based on the balance among exploration and exploitation strategies improving the population diversity at each step during the entire optimization process. However, even when the MFPA outperforms the rest of unimodal algorithms in the multiple circle detection problem. The MFPA presents some drawbacks. At first, the MFPA takes more execution time than the rest of unimodal techniques considered in the experimental study. The larger computational effort in the MFPA is due to its quadratic algorithmic structure. One future work direction of the proposed methodology is to reduce the algorithmic complexity over multiple circle detection problem. For that, some improvements can be implemented in the MFPA such as: improvement of the depuration procedure so the comparison among each memory element can be computed considering another distance metric. The memory structure can be implemented considering a dynamic memory allocation process to accelerate memory access. Also, the fitness-based rules can be substituted considering a fuzzy approach, where membership functions may consider distance and fitness to generate some thumb rules to model uncertainty and improve population diversity.

VI. CONCLUSION

Most of related literature in multi-circle detection problem is based on a unimodal optimization perspective, where

TABLE 7. Synthetic images with different complexities regarding geometric shapes.

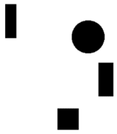
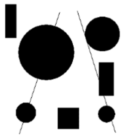
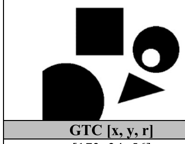
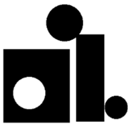
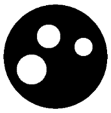
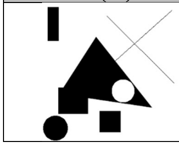
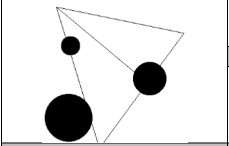
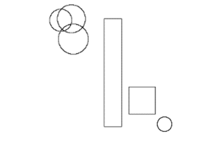
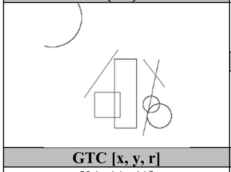
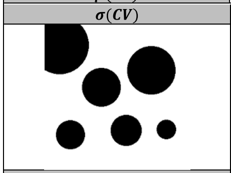
Image	Algorithm									
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.16 s		8.34 s		1.10 s		10.34 s		0.1895	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[54,127,24]	1	3.5695E-03	0.7000	4.3611E-03	1	2.3797E-03	1.1000	5.9483E-03	26.9333	5.8022E-03
$\#GTC = 1$										
Coincidences	1		0.7000		1		1.1000		26.9333	
ECN	1		0.7000		1		1		1.0000	
$\mu(CV)$	1		0.7000		1		1.1000		26.9333	
$\sigma(CV)$	0		0		0		0		0.0000	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.18 s		9.91 s		1.30 s		12.13 s		0.2077s	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[77, 67, 41]	1	9.0913E-03	0.6000	9.5097E-03	1	9.0913E-03	1	1.0573E-02	6.1000	1.0712E-02
[50, 147, 25]	0.8000	3.1338E-03	0	NaN	0.7333	3.1296E-03	1	4.6358E-03	1.0000	4.6623E-03
[164, 37, 15]	0.5667	7.1901E-03	0	NaN	0.4000	7.5130E-03	1	6.0930E-03	0.4333	7.0622E-03
[166, 153, 12]	0.1000	1.4745E-03	0	NaN	0.2000	3.6861E-03	1	2.7830E-03	0.1333	2.6130E-03
$\#GTC = 4$										
Coincidences	2.4667		0.6000		2.3333		4		7.6667	
ECN	2.4667		0.6000		2.3333		4		2.1667	
$\mu(CV)$	0.6167		0.1500		0.5833		1		1.9167	
$\sigma(CV)$	0.5169		0.3000		0.4887		0		2.8866	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.15 s		8.56 s		1.14 s		10.36 s		0.1987	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[173, 34, 56]	0.3333	1.0542E-02	0	NaN	0.2000	1.1459E-02	2.5667	1.0741E-02	0.1000	6.4504E-03
[99, 164, 34]	1	1.3706E-03	0.2333	1.4685E-03	1	1.7133E-03	1.3333	3.5238E-03	14.0000	3.7473E-03
[111, 157, 13]	0.8333	1.4529E-03	0	NaN	0.5333	7.8296E-04	1	2.7386E-03	1.0000	2.7750E-03
$\#GTC = 3$										
Coincidences	2.1667		0.2333		1.7333		4.9000		15.1000	
ECN	2.1333		0.2333		1.7333		3		1.7000	
$\mu(CV)$	0.7222		0.0778		0.5778		1.6333		5.0333	
$\sigma(CV)$	0.4567		0.1347		0.4811		0.9191		7.7980	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.18 s		7.71 s		1.29 s		12.03 s		0.2006	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[16, 82, 27]	1	4.2743E-01	0.3000	4.2332E-01	1	4.2768E-01	1.1667	4.2786E-01	3.6000	4.2670E-01
[144, 57, 23]	1	4.2580E-04	0.2000	2.1290E-03	1	6.3869E-04	1	3.3585E-03	5.5000	3.6101E-03
[165, 183, 17]	0.6333	4.8439E-03	0.0333	0	0.5000	5.1871E-03	1.2667	3.6113E-03	0.8333	4.0769E-03
$\#GTC = 3$										
Coincidences	2.6333		0.5333		2.5000		3.4333		9.9333	
ECN	2.6333		0.5333		2.5000		3		2.5000	
$\mu(CV)$	0.8778		0.1778		0.8333		1.1444		3.3111	
$\sigma(CV)$	0.2117		0.3079		0.2887		0.2502		2.6895	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.21 s		12.04 s		1.50 s		13.72 s		0.2194	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[99, 103, 77]	1	1.3148E-02	0.4333	1.2771E-02	1	1.2857E-02	1.5667	1.1779E-02	22.4667	1.1679E-02
[70, 93, 17]	0.3667	1.5456E-03	0.0333	0	0.1000	2.8336E-03	1	3.9187E-03	0.2333	4.3966E-03
[85, 143, 13]	0.1333	7.2342E-03	0	NaN	0.1333	7.8547E-03	0.9333	4.6148E-03	0.0667	2.8137E-03
[118, 68, 22]	0.5667	3.8375E-03	0	NaN	0.5333	4.7181E-03	1	4.3819E-03	0.7333	4.7826E-03
$\#GTC = 4$										
Coincidences	2.0667		0.4667		1.7667		4.5000		23.5000	
ECN	2.0667		0.4667		1.7667		3.9333		1.7000	
$\mu(CV)$	0.5167		0.1167		0.4417		1.1250		5.8750	
$\sigma(CV)$	0.5272		0.2333		0.5387		0.3105		11.0789	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.16 s		6.58 s		1.14 s		11.03 s		0.1772	

TABLE 7. *(Continued.)* Synthetic images with different complexities regarding geometric shapes.

GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[182, 19, 18]	0.1667	1.0877E-03	0.0333	0	0.1000	1.8129E-03	1.0333	1.0423E-02	0.1333	2.8749E-03
[128, 117, 17]	0.1000	7.6520E-03	0.2333	6.4182E-03	0.0333	0	1.1000	7.3178E-03	0.1333	7.2587E-03
#GTC = 2										
Coincidences	0.2667		0.2667		0.1333		2.1333		0.2667	
ECN	0.2667		0.2667		0.1333		2		0.2667	
$\mu(CV)$	0.1333		0.1333		0.0667		1.0667		0.1333	
$\sigma(CV)$	0.1414		0.1886		0.0943		0.0943		0.1414	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.15 s		8.93 s		1.07 s		9.96 s		0.1640	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[67, 39, 13]	0.3000	1.4135E-03	0.0333	0	0.2333	1.8174E-03	1	5.9510E-03	0.2333	8.9436E-03
[112, 148, 23]	1	1.4691E-03	0	NaN	0.8667	1.4043E-03	1	3.1827E-03	2.9000	3.1986E-03
[166, 36, 33]	1	2.7243E-04	0.2000	2.3254E-03	1	1.2356E-03	1	3.2279E-03	5.1333	3.9927E-03
#GTC = 3										
Coincidences	2.3000		0.2333		2.1000		3		8.2667	
ECN	2.3000		0.2333		2.1000		3		2.1333	
$\mu(CV)$	0.7667		0.0778		0.7000		1		2.7556	
$\sigma(CV)$	0.4041		0.1347		0.4426		0		2.8197	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.10 s		7.03 s		0.72 s		7.53 s		0.1133	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[30, 30, 15]	1.2000	6.0281E-02	0.1333	9.4076E-02	1	6.3821E-02	4.8333	7.8771E-02	0.8000	6.7499E-02
[28, 45, 19]	0.7667	4.6982E-02	0.0667	3.9715E-02	0.4333	4.7407E-02	3.5667	4.7717E-02	0.4667	4.7892E-02
[56, 47, 21]	1	3.3826E-02	0.1667	3.7618E-02	0.6667	3.2043E-02	3.5667	3.6262E-02	1.2000	3.1656E-02
[173, 173, 10]	0.4000	1.1315E-02	0	NaN	0.2667	1.1198E-02	1.7333	9.8089E-03	0.5333	9.4980E-03
#GTC = 4										
Coincidences	3.3667		0.3667		2.3667		13.7000		3.0000	
ECN	2.4000		0.3000		2		4		2.0667	
$\mu(CV)$	0.8417		0.0917		0.5917		3.4250		0.7500	
$\sigma(CV)$	0.8094		0.1833		0.6709		1.4933		0.8326	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.17 s		8.15 s		1.28 s		11.54 s		0.1971	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[21, 11, 41]	0	NaN	0	NaN	0	NaN	1.2000	8.4487E-02	0.0000	NaN
[141, 148, 11]	0.1667	1.2344E-02	0.0333	1.8278E-02	0.3333	1.3003E-02	1.1000	1.3256E-02	0.0333	1.2521E-02
[157, 159, 17]	0.3333	1.3179E-02	0.6333	1.1728E-02	0.2667	1.0485E-02	2.3333	1.0421E-02	0.2333	1.0226E-02
#GTC = 3										
Coincidences	0.5000		0.6667		0.6000		4.6333		0.2667	
ECN	0.5000		0.2667		0.6000		2.8667		0.2000	
$\mu(CV)$	0.1667		0.2222		0.2000		1.5444		0.0889	
$\sigma(CV)$	0.2502		0.3849		0.3079		0.8193		0.1540	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.18 s		8.67 s		1.30 s		12.56 s		0.1988	
GTC [x, y, r]	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*	*VC*	*NCD*
[29, 21, 40]	0.1000	2.8317E-02	0	NaN	0	NaN	1.0333	2.2059E-02	0.0000	NaN
[87, 79, 27]	0.6333	8.8465E-03	0.0667	8.2934E-03	0.5000	9.0613E-03	1	6.5636E-03	0.7667	7.0863E-03
[64, 147, 33]	0.9333	1.3091E-03	0.5000	1.7978E-03	1	1.0182E-03	1	3.4351E-03	2.8000	4.0953E-03
[153, 36, 20]	0.2667	6.1760E-03	0.1333	4.7335E-03	0.4333	4.9727E-03	1	4.6064E-03	0.2333	5.7642E-03
[147, 113, 21]	0.3000	2.9773E-03	0.1333	6.3399E-03	0.2333	4.2570E-03	1	4.5327E-03	0.1333	4.7214E-03
[145, 168, 13]	0.1333	0	0	NaN	0.1333	3.8396E-03	1	2.0074E-03	0.0333	3.3775E-03
#GTC = 6										
Coincidences	2.3667		0.8333		2.3000		6.0333		3.9667	
ECN	2.3667		0.8333		2.3000		5.9333		1.8667	
$\mu(CV)$	0.3944		0.1389		0.3833		1.0056		0.6611	
$\sigma(CV)$	0.5093		0.3202		0.5063		0.0680		1.2172	

evolutionary algorithms present an interesting alternative to automatic circle detection task. However, under a unimodal approach, each circle is detected over a single execution of the optimization algorithm. Under such circumstances, multi-circle detection yields to multiple executions of the optimization process. Since real-world applications are modeled as multimodal optimization surfaces, the extensive necessity of real multimodal optimization algorithms

TABLE 8. Real photos containing multiple circle shapes.

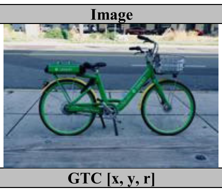
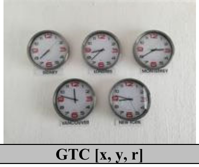
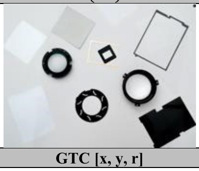
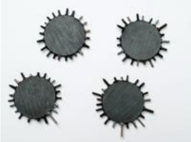
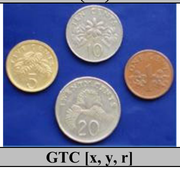
Image	Algorithm									
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.12 s		6.27 s		0.85 s		8.70 s		0.1292	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[80, 61, 28]	0.1667	1.4575E-02	0.0333	1.3543E-02	0.1333	1.4304E-02	1	1.1973E-02	0.2000	1.0930E-02
[81, 61, 23]	0.0333	1.3601E-02	0	NaN	0	NaN	0.9333	1.5937E-02	0.0000	NaN
[80, 155, 26]	0.2000	1.6567E-02	0.3000	1.4762E-02	0.1333	2.2602E-02	1.6667	2.5722E-02	0.1667	1.4438E-02
[80, 155, 21]	0.2000	8.6388E-03	0.3000	1.2791E-02	0.1000	1.5239E-02	1.1667	1.5006E-02	0.0667	1.1477E-02
#GTC = 4										
Coincidences	0.6000		0.6333		0.3667		4.7667		0.4333	
ECN	0.6000		0.5667		0.3667		3.8333		0.4333	
$\boldsymbol{\mu(CV)}$	0.1500		0.1583		0.0917		1.1917		0.1083	
$\boldsymbol{\sigma(CV)}$	0.2436		0.2382		0.1692		0.4893		0.2167	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.15 s		7.31 s		1.09 s		10.46 s		0.1612	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[55, 48, 22]	0.0333	1.8549E-02	0.0667	1.5832E-02	0.0667	1.5832E-02	1.0667	1.3529E-02	0.1000	1.8293E-02
[55, 103, 21]	0.1000	7.9469E-03	0	NaN	0.0333	1.1920E-02	1	1.2064E-02	0.0333	1.1372E-02
[55, 157, 21]	0.1333	1.0452E-02	0.2667	9.1451E-03	0	NaN	1	9.9183E-03	0.0000	NaN
[106, 76, 22]	0	NaN	0	NaN	0	NaN	0.8333	1.0963E-02	0.0667	1.2237E-02
[106, 132, 22]	0.0667	9.2149E-03	0.1667	8.1711E-03	0	NaN	1	7.8112E-03	0.0000	NaN
#GTC = 5										
Coincidences	0.3333		0.5000		0.1000		4.9000		0.2000	
ECN	0.3333		0.5000		0.1000		4.8333		0.2000	
$\boldsymbol{\mu(CV)}$	0.0667		0.1000		0.0200		0.9800		0.0400	
$\boldsymbol{\sigma(CV)}$	0.1491		0.2236		0.0447		0.0981		0.0894	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.13 s		7.16 s		1.00 s		9.76 s		0.1513	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[64, 58, 16]	0.1000	2.3543E-02	0.0667	2.6671E-02	0.1000	1.9663E-02	1	2.6293E-02	0.0667	2.0861E-02
[108, 93, 17]	0.0667	1.3823E-02	0.0667	9.8528E-03	0.2000	1.3918E-02	0.8333	1.2534E-02	0.2000	1.1728E-02
[107, 93, 10]	0.0667	3.5182E-03	0.0333	9.9509E-03	0	NaN	0.3333	7.3372E-03	0.0000	NaN
[89, 142, 19]	0.1333	1.0679E-02	0.2000	1.3182E-02	0.0333	1.3258E-02	1.0333	3.7786E-02	0.0667	1.3592E-02
[88, 142, 13]	0.0333	8.4399E-03	0.1333	8.4399E-03	0.0333	1.0337E-02	0.5000	8.3045E-03	0.0000	NaN
#GTC = 5										
Coincidences	0.4000		3.7333		0.5000		0.3667		0.3333	
ECN	0.4000		3.6667		0.5000		0.3667		0.3000	
$\boldsymbol{\mu(CV)}$	0.0800		0.7467		0.1000		0.0733		0.0667	
$\boldsymbol{\sigma(CV)}$	0.1673		0.4379		0.2005		0.1293		0.1375	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.15 s		7.96 s		1.06 s		10.38 s		0.1584	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[40, 68, 23]	0.7000	2.4072E-02	0.4000	2.4887E-02	0.6667	2.2970E-02	1.1000	2.8978E-02	0.7333	2.4422E-02
[45, 149, 22]	0.0667	1.7334E-02	0.0333	1.5583E-02	0.1000	1.7101E-02	1	1.5589E-02	0.1000	1.6981E-02
[104, 38, 23]	0.1000	1.8916E-02	0	NaN	0.1000	2.1168E-02	1.0333	2.0594E-02	0.1000	1.5086E-02
[109, 129, 24]	0	NaN	0	NaN	0	NaN	0.7000	1.1841E-02	0.0000	NaN
#GTC = 4										
Coincidences	0.8667		0.4333		0.8667		3.8333		0.9333	
ECN	0.8667		0.4333		0.8667		3.6667		0.7667	
$\boldsymbol{\mu(CV)}$	0.2167		0.1083		0.2167		0.9583		0.2333	
$\boldsymbol{\sigma(CV)}$	0.3911		0.2167		0.3911		0.2316		0.4244	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		Execution time		Execution time		Execution time		Execution time	
	0.15 s		8.15 s		1.14 s		10.99 s		0.1700	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[73, 35, 30]	0.7333	1.6159E-02	0.1000	1.6380E-02	0.6667	1.7088E-02	1	1.7725E-02	1.0333	1.7190E-02
[36, 107, 33]	0.9667	1.3520E-02	0.4000	1.2533E-02	0.7333	1.3678E-02	1	1.4567E-02	1.4667	1.4587E-02
[82, 169, 28]	0.8000	2.6327E-03	0	NaN	0.6667	2.8893E-03	1.1333	4.7658E-03	1.7000	4.0414E-03
[123, 99, 38]	0.9667	7.7407E-03	0.2333	7.3794E-03	0.8667	7.9758E-03	1	7.6077E-03	2.4000	7.2733E-03
#GTC = 4										
Coincidences	3.4667		0.7333		2.9333		4.1333		6.6000	

TABLE 8. *(Continued.)* Real photos containing multiple circle shapes.

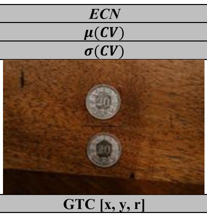
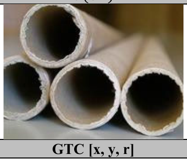
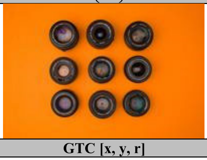
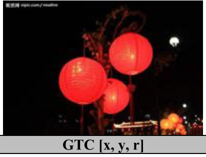
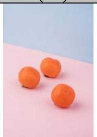
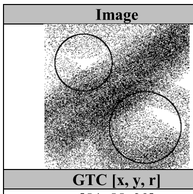
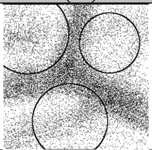
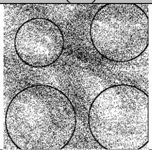
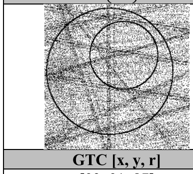
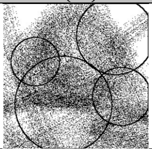
ECN	3.4667		0.7333		2.9333		4		3.1333	
μ(CV)	0.8667		0.1833		0.7333		1.0333		1.6500	
σ(CV)	0.2192		0.3385		0.3155		0.0667		1.3458	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.13 s		7.45 s		1.01 s		9.23 s		0.1479	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[45, 101, 16]	0.9333	1.7625E-02	0.1667	1.8543E-02	0.8333	1.7340E-02	1	1.7574E-02	1.6000	1.8596E-02
[92, 102, 16]	0.9667	1.0164E-02	0.1667	1.1415E-02	0.9333	1.0753E-02	1	1.0880E-02	1.7333	1.0818E-02
#GTC = 2										
Coincidences	1.9000		0.3333		1.7667		2		3.3333	
ECN	1.9000		0.3333		1.7667		2		1.5333	
μ(CV)	0.9500		0.1667		0.8833		1		1.6667	
σ(CV)	0.0707		0.0943		0.1179		0		1.1785	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.12 s		7.78 s		0.93 s		9.56 s		0.1389	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[34, 57, 37]	0.0333	2.9427E-02	0.0333	3.2236E-02	0.0667	3.0831E-02	1.0667	2.6361E-02	0.0333	1.9522E-02
[34, 57, 30]	0.1333	3.3879E-02	0.0667	4.3417E-02	0.0667	3.0700E-02	1	4.1327E-02	0.0333	3.5093E-02
[94, 17, 27]	0	NaN	0	NaN	0	NaN	0.9000	1.9128E-02	0.0000	NaN
[101, 93, 39]	0.1000	1.2780E-02	0.0333	9.9085E-03	0.2000	1.1269E-02	1	7.2475E-03	0.1333	7.3308E-03
[102, 92, 33]	0.5000	8.9895E-03	0.1667	1.0911E-02	0.3333	1.1717E-02	1.0333	8.7027E-03	0.3667	9.3928E-03
[109, 169, 38]	0	NaN	0	NaN	0	NaN	0	NaN	0.0000	NaN
[110, 169, 32]	0.1000	8.5486E-03	0.0333	9.7958E-03	0.1667	9.3419E-03	1	1.0799E-02	0.2333	1.1643E-02
#GTC = 7										
Coincidences	0.8667		0.3333		0.8333		6		0.8000	
ECN	0.8667		0.3333		0.8333		5.9000		0.7333	
μ(CV)	0.1238		0.0476		0.1190		0.8571		0.1143	
σ(CV)	0.2829		0.1171		0.2593		0.4089		0.2398	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.14 s		7.37 s		1.08 s		10.40 s		0.1595	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[33, 63, 15]	0	NaN	0.0333	2.7516E-02	0.0333	1.3758E-02	0.9667	2.8840E-02	0.0000	NaN
[32, 99, 15]	0.0333	2.6907E-02	0.1000	2.1221E-02	0.0667	1.8378E-02	0.7667	2.0323E-02	0.0333	1.9962E-02
[34, 136, 15]	0.1333	1.0031E-02	0	NaN	0.1000	1.0995E-02	0.9667	1.0844E-02	0.1000	1.2956E-02
[68, 62, 14]	0.1000	1.3710E-02	0	NaN	0.0333	1.5193E-02	0.8000	1.3722E-02	0.0000	NaN
[67, 100, 14]	0	NaN	0.0667	9.9611E-03	0.0333	0	0.7000	1.2130E-02	0.0333	1.2215E-02
[67, 135, 14]	0	NaN	0.0333	6.6067E-03	0.1333	8.9426E-03	0.9333	8.8296E-03	0.1000	8.6600E-03
[101, 63, 14]	0.0667	1.5227E-02	0	NaN	0.0667	1.9546E-02	0.7333	1.4629E-02	0.1333	1.7598E-02
[102, 101, 14]	0.1000	1.3605E-02	0.0667	1.2655E-02	0.1333	1.3025E-02	0.7667	1.3595E-02	0.1000	1.4646E-02
[101, 134, 14]	0.0333	8.3988E-03	0.0333	8.3988E-03	0.1000	8.3988E-03	0.7667	1.0035E-02	0.0667	1.0778E-02
#GTC = 9										
Coincidences	0.4667		0.3333		0.7000		7.4000		0.5667	
ECN	0.4667		0.3333		0.7000		7.4000		0.5667	
μ(CV)	0.0519		0.0370		0.0778		0.8222		0.0630	
σ(CV)	0.1330		0.0944		0.1941		0.3432		0.1663	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.13 s		7.14 s		0.97 s		8.37 s		0.1431	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[81, 81, 24]	0.6667	8.3243E-03	0	NaN	0.5333	7.8269E-03	1	6.8055E-03	0.5333	7.4945E-03
[55, 129, 21]	1	1.1282E-02	0.5000	1.1022E-02	1	9.9687E-03	1	1.1761E-02	2.7333	1.0703E-02
#GTC = 2										
Coincidences	1.6667		0.5000		1.5333		2		3.2667	
ECN	1.6667		0.5000		1.5333		2		1.3000	
μ(CV)	0.8333		0.2500		0.7667		1		1.6333	
σ(CV)	0.2357		0.3536		0.3300		0		1.6499	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.13 s		6.91 s		0.99 s		9.42 s		0.1495	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[97, 72, 15]	0.5667	1.1771E-02	0	NaN	0.2333	1.1618E-02	1	1.2380E-02	0.1667	1.3521E-02
[112, 40, 16]	0.5000	1.1041E-02	0	NaN	0.5000	1.1394E-02	1	1.1505E-02	0.6667	1.1819E-02
[139, 90, 18]	0.9667	1.2066E-03	0.4000	4.0024E-03	0.9333	1.6785E-03	1.0667	3.3352E-03	3.1000	5.1918E-03
#GTC = 3										
Coincidences	2.0333		0.4000		1.6667		3.0667		3.9333	
ECN	2.0333		0.4000		1.6667		3		1.6000	
μ(CV)	0.6778		0.1333		0.5556		1.0222		1.3111	
σ(CV)	0.4426		0.2309		0.4811		0.0385		1.6659	

TABLE 9. Synthetic images with different complexities containing noise.

Image	Algorithm									
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.17 s		9.76 s		1.13 s		12.17 s		0.1813	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[54, 55, 39]	0	NaN	0.4667	3.1090E-02	0.0333	3.4729E-02	0.9667	2.9096E-02	0.0333	3.6507E-02
[143, 140, 49]	0	NaN	0.3667	1.1648E-02	0.0333	9.7071E-03	1.9000	2.3460E-02	0.0667	1.7257E-01
#GTC = 2										
Coincidences	0		0.8333		0.0667		2.8667		0.1000	
ECN	0		0.4667		0.0667		1.8000		0.1000	
$\boldsymbol{\mu(CV)}$	0		0.4167		0.0333		1.4333		0.0500	
$\boldsymbol{\sigma(CV)}$	0		0.5893		0.0471		0.7542		0.0707	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.16 s		9.09 s		0.99 s		11.57 s		0.1759	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[41, 35, 56]	0	NaN	0.0333	3.1513E-02	0.0667	3.0140E-02	1.7667	3.4052E-02	0.0000	NaN
[54 147, 41]	0.8000	1.6360E-02	1.0333	1.6487E-02	0.5333	1.6212E-02	2.4333	1.5154E-02	0.3667	1.6006E-02
[162, 93, 52]	0.5000	1.4294E-02	0.1000	1.5200E-02	0.3667	1.3684E-02	2.5000	1.2476E-02	0.4667	1.4243E-02
#GTC = 3										
Coincidences	1.3000		1.1667		0.9667		6.7000		0.8333	
ECN	1.0667		0.4667		0.9000		3		0.6333	
$\boldsymbol{\mu(CV)}$	0.4333		0.3889		0.3222		2.2333		0.2778	
$\boldsymbol{\sigma(CV)}$	0.5743		0.6736		0.4375		0.7104		0.3922	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.16 s		8.54 s		0.97 s		11.09 s		0.1753	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[54, 51, 33]	0	NaN	0	NaN	0	NaN	0.8667	2.9817E-02	0.0000	NaN
[39, 163, 41]	0	NaN	0.1000	1.3161E-02	0	NaN	1.5667	2.3382E-02	0.0000	NaN
[161, 53, 48]	0.0333	1.2693E-02	0.3333	1.2278E-02	0.0333	1.2693E-02	1.7667	1.3208E-02	0.0333	1.4469E-02
[159, 165, 47]	0.1000	1.0952E-02	0.5333	1.2952E-02	0.0667	1.3330E-02	1.7667	1.0723E-02	0.0000	NaN
#GTC = 4										
Coincidences	0.1333		0.9667		0.1000		5.9667		0.0333	
ECN	0.1333		0.5333		0.1000		3.6333		0.0333	
$\boldsymbol{\mu(CV)}$	0.0333		0.2417		0.0250		1.4917		0.0083	
$\boldsymbol{\sigma(CV)}$	0.0667		0.4316		0.0500		0.7419		0.0167	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.17 s		10.28 s		1.04 s		9.66 s		0.1855	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[93, 91, 87]	0.5000	7.1844E-02	0.6333	1.5080E-02	0.2333	1.5034E-02	2.4000	2.5854E-02	0.2333	1.5532E-02
[71, 111, 47]	0.0667	2.0878E-02	0.1000	1.7750E-02	0.1000	1.7241E-02	2.0333	4.1086E-02	0.0667	1.4818E-02
#GTC = 2										
Coincidences	0.5667		0.7333		0.3333		4.4333		0.3000	
ECN	0.5000		0.5000		0.3333		2		0.2667	
$\boldsymbol{\mu(CV)}$	0.2833		0.3667		0.1667		2.2167		0.1500	
$\boldsymbol{\sigma(CV)}$	0.4007		0.5185		0.2357		0.3536		0.2121	
	DE		GWO		HS		MFPA		MPSO	
	Execution time		**Execution time**		**Execution time**		**Execution time**		**Execution time**	
	0.17 s		9.65 s		1.05 s		9.39 s		0.1836	
GTC [x, y, r]	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***	***VC***	***NCD***
[45, 155, 53]	0.0667	1.6039E-02	0.1667	1.3966E-02	0.1000	2.2259E-01	1.8000	1.4107E-02	0.0333	1.2829E-02
[81, 45, 33]	0	NaN	0	NaN	0	NaN	0.9333	2.8035E-02	0.0000	NaN
[129, 164, 40]	0	NaN	0	NaN	0	NaN	1.0333	2.0926E-02	0.0333	1.1781E-02
[140, 84, 67]	0.1333	1.5917E-02	0.0667	1.6999E-02	0.1000	1.3343E-02	1.7000	2.5740E-02	0.0000	NaN
#GTC = 4										
Coincidences	0.2000		0.2333		0.2000		5.4667		0.0667	
ECN	0.2000		0.2000		0.2000		3.7333		0.0667	
$\boldsymbol{\mu(CV)}$	0.0500		0.0583		0.0500		1.3667		0.0167	
$\boldsymbol{\sigma(CV)}$	0.1000		0.1167		0.0859		0.6386		0.0333	

is required. In this paper, the multi-circle detection problem has been reformulated as a real multimodal optimization problem. In the proposed approach, a single run of a multimodal optimization algorithm yields to multiple optima solutions. The proposed approach considers the Multimodal Flower Pollination Algorithm (MFPA) to obtain multiple circles on an image in a single run. According to the experimental study, MFPA presents outstanding performance in comparison with traditional unimodal optimization algorithms commonly used in multi-circle detection problem. The proposed method finds in most of the cases all the circles in the analyzed images in a single execution. Furthermore, the experimental results suggest that MFPA considers high tolerance to the presence of noise, outperforming the rest of the algorithms in noisy and real images. Also, the proposed method is able to remove redundant solutions.

This research work considers how multimodal approaches can be a viable alternative to solve practical problems that require multiple solutions to being localized despite being a relatively little explored area in metaheuristics. However, it is possible that the adaptation of problems to the multimodal approach could not being as trivial as applying the algorithm directly on the search space of the problem and thus, it could be necessary a deep analysis of the problem and the algorithm for it to work properly. For example, in the multi-circle detection problem, it is necessary to change how solutions are expressed in the memory capture phase and the depuration procedure to make sense with the main goal of the multimodal optimization process. Possible future work could consist in adapting other real-world problems (not necessarily related to image analysis) to be solved using multimodal approaches since, as it was shown, multimodal techniques can be a good alternative and even a more fitting approach to tackle certain types of problems that otherwise, could require a more complex or expensive process for being solve

APPENDIX

In this Appendix, the numerical results for Section V. are presented.

See Table 7–9.

REFERENCES

[1] L. da Fona Costa and R. M. Cesar Jr., *Shape Classification and Analysis: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2009. Accessed: Dec. 12, 2022. [Online]. Available: https://www.routledge.com/Shape-Classification-and-Analysis-Theory-and-Practice-Second-Edition/Costa-Cesar-Jr/p/book/9780849379291

[2] R. Manish, A. Venkatesh, and S. D. Ashok, "Machine vision based image processing techniques for surface finish and defect inspection in a grinding process," *Mater. Today, Proc.*, vol. 5, no. 5, pp. 12792–12802, 2018, doi: 10.1016/J.MATPR.2018.02.263.

[3] D. Manolakis, D. Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *LINCOLN Lab. J.*, vol. 14, no. 1, pp. 79–116, 2003.

[4] S. B. Dworkin and T. J. Nye, "Image processing for machine vision measurement of hot formed parts," *J. Mater. Process. Technol.*, vol. 174, nos. 1–3, pp. 1–6, May 2006, doi: 10.1016/J.JMATPROTEC.2004.10.019.

[5] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of Hough transform methods for circle finding," *Image Vis. Comput.*, vol. 8, no. 1, pp. 71–77, Feb. 1990, doi: 10.1016/0262-8856(90)90059-E.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.

[7] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *Pattern Recognit. Lett.*, vol. 11, no. 5, pp. 331–338, 1990, doi: 10.1016/0167-8655(90)90042-Z.

[8] J. H. Han, L. T. Kóczy, and T. Poston, "Fuzzy Hough transform," *Pattern Recognit. Lett.*, vol. 15, pp. 649–658, Jul. 1994, doi: 10.1016/0167-8655(94)90068-X.

[9] C. Cartis, L. Roberts, and O. Sheridan-Methven, "Escaping local minima with local derivative-free methods: A numerical investigation," *Optimization*, vol. 71, no. 8, pp. 2343–2373, Aug. 2022, doi: 10.1080/02331934.2021.1883015.

[10] K. Hansel, J. Moos, and C. Derstroff, "Benchmarking the natural gradient in policy gradient methods and evolution strategies," in *Reinforcement Learning Algorithms: Analysis and Applications*, vol. 883, 2021, pp. 69–84, doi: 10.1007/978-3-030-41188-6_7.

[11] C. Cartis, T. Ferguson, and L. Roberts, "Scalable derivative-free optimization for nonlinear least-squares problems," 2020, *arXiv:2007.13243*.

[12] C. Akinlar and C. Topal, "EDCircles: A real-time circle detector with a false detection control," *Pattern Recognit.*, vol. 46, no. 3, pp. 725–740, Mar. 2013, doi: 10.1016/J.PATCOG.2012.09.020.

[13] L. Huang, S. Ding, S. Yu, J. Wang, and K. Lu, "Chaos-enhanced Cuckoo search optimization algorithms for global optimization," *Appl. Math. Model.*, vol. 40, nos. 5–6, pp. 3860–3875, Mar. 2016, doi: 10.1016/J.APM.2015.10.052.

[14] S. Dasgupta, S. Das, A. Biswas, and A. Abraham, "Automatic circle detection on digital images with an adaptive bacterial foraging algorithm," *Soft Comput.*, vol. 14, no. 11, pp. 1151–1164, Sep. 2010, doi: 10.1007/s00500-009-0508-z.

[15] E. Cuevas, D. Zaldivar, M. Pérez-Cisneros, and M. Ramírez-Ortegón, "Circle detection using discrete differential evolution optimization," *Pattern Anal. Appl.*, vol. 14, no. 1, pp. 93–107, Feb. 2011, doi: 10.1007/s10044-010-0183-9.

[16] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: Univ. Michigan Press, 1975.

[17] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Comput. Eng. Dept., Eng. Fac., Erciyes Univ., Kayseri, Turkey, Tech. Rep., 2005.

[18] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.*, 2013, pp. 240–249, doi: 10.1007/978-3-642-32894-7_27.

[19] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: 10.1016/j.ins.2009.03.004.

[20] I. Birbil and S.-C. Fang, "An electromagnetism-like mechanism for global optimization," *J. Global Optim.*, vol. 25, no. 3, pp. 263–282, 2003.

[21] T. S. L. V. Ayyarao, N. S. S. Ramakrishna, R. M. Elavarasan, N. Polumahanthi, M. Rambabu, G. Saini, B. Khan, and B. Alatas, "War strategy optimization algorithm: A new effective metaheuristic algorithm for global optimization," *IEEE Access*, vol. 10, pp. 25073–25105, 2022, doi: 10.1109/ACCESS.2022.3153493.

[22] A. Lopez-Martinez and F. J. Cuevas, "Automatic circle detection on images using the teaching learning based optimization algorithm and gradient analysis," *Int. J. Speech Technol.*, vol. 49, no. 5, pp. 2001–2016, Dec. 2018, doi: 10.1007/S10489-018-1372-2.

[23] V. Rajinikanth, N. Dey, E. Kavallieratou, and H. Lin, "Firefly algorithm-based Kapur's thresholding and Hough transform to extract leukocyte section from hematological images," in *Applications of Firefly Algorithm and its Variants*, 2020, pp. 221–235, doi: 10.1007/978-981-15-0306-1_10.

[24] J. Pruthi, K. Khanna, and S. Arora, "Optic cup segmentation from retinal fundus images using glowworm swarm optimization for glaucoma detection," *Biomed. Signal Process. Control*, vol. 60, Jul. 2020, Art. no. 102004, doi: 10.1016/J.BSPC.2020.102004.

[25] V. Ayala-Ramirez, C. H. Garcia-Capulin, A. Perez-Garcia, and R. E. Sanchez-Yanez, "Circle detection on images using genetic algorithms," *Pattern Recognit. Lett.*, vol. 27, no. 6, pp. 652–657, Apr. 2006, doi: 10.1016/J.PATREC.2005.10.003.

[26] E. Cuevas, A. Trujillo, M. A. Navarro, and P. Diaz, "Comparison of recent metaheuristic algorithms for shape detection in images," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 1059–1071, Aug. 2020, doi: 10.2991/IJCIS.D.200729.001.

[27] J. Gálvez, E. Cuevas, and O. Avalos, "Flower pollination algorithm for multimodal optimization," *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, p. 627, 2017, doi: 10.2991/ijcis.2017.10.1.42.

[28] M. L. V. Pitteway, "Algorithm for drawing ellipses or hyperbolae with a digital plotter," *Comput. J.*, vol. 10, no. 3, pp. 282–289, Mar. 1967, doi: 10.1093/COMJNL/10.3.282.

[29] J. Bresenham, "A linear algorithm for incremental digital display of circular arcs," *Commun. ACM*, vol. 20, no. 2, pp. 100–106, Feb. 1977, doi: 10.1145/359423.359432.

[30] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep., 1995.

[31] Z. Woo Geem, J. Hoon Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001, doi: 10.1177/003754970107600201.

[32] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

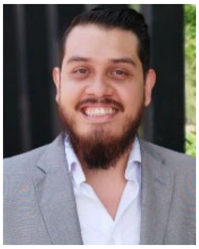
[33] J. Barrera and C. A. C. Coello, "A review of particle swarm optimization methods used for multimodal optimization," in *Innovations in Swarm Intelligence*, vol. 248, 2009, pp. 9–37, doi: 10.1007/978-3-642-04225-6_2/COVER.

[34] E. Cuevas, N. Ortega-Sánchez, D. Zaldivar, and M. Pérez-Cisneros, "Circle detection by harmony search optimization," *J. Intell. Robot. Syst.*, vol. 66, no. 3, pp. 359–376, 2012, doi: 10.1007/S10846-011-9611-3/METRICS.

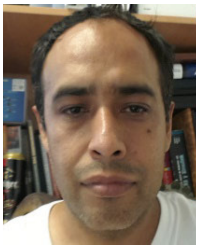
[35] A. Cetin, S. Rezai, and T. Akan, "Multi-circle detection using multimodal optimization," in *Engineering Applications of Modern Metaheuristics*, vol. 1069, 2023, pp. 193–209, doi: 10.1007/978-3-031-16832-1_11/COVER.

[36] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, Feb. 2019, doi: 10.1016/j.swevo.2018.06.010.

[37] L. Liu, J. Huo, F. Xue, and Y. Dai, "Harmony search method with global sharing factor based on natural number coding for vehicle routing problem," *Information*, vol. 11, no. 2, p. 86, Feb. 2020, doi: 10.3390/info11020086.



JORGE GÁLVEZ received the B.S. degree in informatics, in 2011, the M.Sc. degree in electronics and computing, in 2015, and the Ph.D. degree in electronics and computer science from Universidad de Guadalajara, in 2019. He is currently a Professor with the Department of Computer Science, Universidad de Guadalajara. His research interests include the design and applications of metaheuristic algorithms, machine learning, complex systems, and robotics. Since 2019, he has been a member of the Mexican National Research System (SNI I).



ERIK CUEVAS received the B.S. degree (Hons.) in electronics and communications engineering from the University of Guadalajara, Mexico, in 1995, the M.Sc. degree in industrial electronics from ITESO, Mexico, in 2000, and the Ph.D. degree from Freie Universität Berlin, Germany, in 2006. Since 2006, he has been with the University of Guadalajara, where he is currently a full-time Professor with the Department of Computer Science and a Professor with the Department of Electronics. His current research interests include metaheuristics and computer vision. Since 2008, he has been a member of the Mexican National Research System (SNI III). He serves as an Associate Editor for *Artificial Intelligence Review*, *Expert System with Applications*, *ISA Transactions*, and *Applied Soft Computing*.

DAVID ELIEL BOCANEGRA MICHEL received the B.S. degree in computer engineering and the M.S. degree in electronics and computer science engineering from Universidad de Guadalajara, Jalisco, Mexico, in 2017 and 2022, respectively. He was a System Administrator and a Programmer Analyst with Jesco Solutions, from 2017 to 2019. Currently, he is a Developer of the Federal Electricity Commission (CFE) as part of the Innovation Group. He has contribute to the book *Metaheuristics in Machine Learning: Theory and Applications* (Springer, 2021), as coauthor of the Chapter Genetic Operators and Their Impact on the Training of Deep Neural Networks.

ALMA RODRÍGUEZ received the Ph.D. degree in electronics and computer science and engineering from the University of Guadalajara, Mexico. She is currently a Professor and a Researcher with the Department of Electronics and Computer Science, University of Guadalajara. Her scientific production currently includes three books and more than 25 published research articles in international journals. Her research interests include metaheuristic algorithms and artificial intelligence. She received the L'Oréal-UNESCO For Women in Science International Award, in 2022.

MARCO ANTONIO PÉREZ-CISNEROS received the B.Eng. degree in communications and electronics engineering from the University of Guadalajara, Mexico, the M.Eng. degree from ITESO University, Mexico, and the Ph.D. degree from the University of Manchester, U.K. He serves as a Professor with the Electro-Photonics Department and has been appointed as the Chancellor of the Exact Sciences and Engineering Centre, University of Guadalajara. Since 2018, he has been a member of the Mexican National Science Academy. He is a member of the National Research System in Mexico. He is regular member of the IET U.K. He also serves as an Associated Editor for the journal of IEEE Communication Letters.